A COMMERCIAL AND MILITARIZED 737 SYSTEM AND SUBSYSTEM LEVEL ANALYSIS: A SYSTEMS ENGINEERING MODELING AND SIMULATION APPROACH

BY

CRAIG MICHAEL PAUGA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Advisor:

Professor Steven J. D'Urso

Abstract

This reading focuses on using a systems engineering modeling approach to determine the performance deviation between a commercial and militarized versions of the Boeing 737. In systems engineering complex systems are designed around a concept of operations or a specific mission that the system was originally designed to accomplish. In the cases of the militarized variations of the Boeing 737, the Airborne Early Warning & Control (AEW&C) and the Airborne Warning and Control System (AWACS), these systems were used for different operational purposes than its original concept of operations (ConOps). This study will use systems engineering principles to breakdown the variances between these military variations and its original design in the Boeing 737. Systems engineering modeling focuses on breaking down the product by its systems followed by modeling the behavior and interaction of these systems. Physical models of both the militarized and commercial were built using the Aircraft Synthesis (ACS) tool. This tool, used independently, is capable of modeling and simulating external performances of the two aircraft variations. The internal performances of the two systems will be modeled within the Rolls-Royce PowerFlow toolset. The PowerFlow toolset gives a high fidelity simulation of internal performance of aircraft subsystems. Using this tool, it will be possible to determine power loads required for the various subsystems. The PowerFlow tool will give an idea the difference between power load requirements between the two aircraft variations. Internal performance can have a large effect on an aircraft's mission capabilities. For this reason, the results from the ACS tool and the PowerFlow tool will then be coupled to determine the impact power draw can have on the overall system.

Table of Contents

List of Figures

List of Tables

Terms/Acronyms

R/C: Rate of Climb

ACS: Aircraft Synthesis Tool

SOS: System of Systems (Level 0)

T-C: Thickness to Chord

AR: Aspect Ratio

C_f: Coefficient of Friction

Re: Reynold's Number

M: Mach

D: Drag

L: Lift

GUI: Graphic User Interface

EWSP: Electronic Warfare Self-Protection

SLTA: Small Laser Transmitter Assembly

Mini Pointer Tracking Assembly: MPTA

Standard Missile Approach Warner: SMAW

Multi Image Multi Spectral Missile Approach Warner: MIMS MAW

Cabin Interface Unit: CIU

V_q: Quadrature axis synchronous machine terminal voltage

V_d: Direct synchronous machine terminal voltage

R_{TL}: Transmission per-unit resistance

X_{TL}: Transmission per-unit reactance

R_S: Synchronous machine per-unit stator resistance

γ: Per-unit electrical frequency

X_d: Direct per-unit reactance

X_q: Quadrature per-unit reactance

X_d': Direct per-unit transient reactance

X_q': Quadrature per-unit transient reactance

X_d'': Direct per-unit subtransient reactance

X_q'': Quadrature per-unit subtransient reactance

T_{do}': Direct field winding per-unit transient time constant

T_{qo}': Quadrature field winding per-unit transient time constant

T_{do}'': Direct field winding per-unit subtransient time constant

T_{qo}'': Quadrature field winding per-unit subtransient time constant

V_{dqo}: Synchronous machine line voltage

P: Power

$\tilde{P}$: Pressure of fluid

I: Current

S_b: Base power

T_g: Per-unit torque to generator

I_d: Direct frame line current

I_q: Quadrature frame line current

E_d: Direct frame line voltage

E_q: Quadrature frame line voltage

C_p: Specific heat constant at constant pressure

T_{bleed}: Temperature of bleed air

T_{cabin}: Temperature of cabin

1. System Level Overview

1.1. System Level Comparison

It is pivotal to initially establish that for the bulk of this study the AEW&C and AWACS aircraft will be modeled as the same military variation of the commercial 737. This is due to many specifications of the subsystems being very sparse in literature. The commercial variations model will sometime be referred to as the 737-C and the military variation will be referred to as the 737-M.

In systems engineering the primary step in order to analyze complex systems is to break the system-of-systems into its lower level systems. When breaking down a complex system one must begin at the system-of systems or the level 0 system. For this study, aircraft systems will be the assumed level 0 system. A commercialized 737 aircraft and a militarized variation have the same Level 0 system of systems as well as the same Level 1 systems. The diagram below shows the Level 0 and Level 1 systems hierarchy.

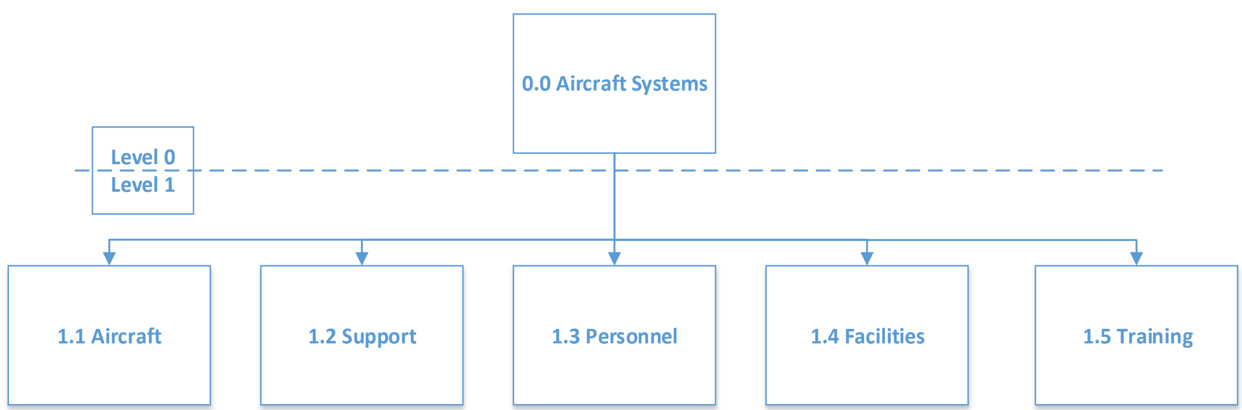

Figure 1. Level 0 and Level 1 System Hierarchy

To limit the scope of this study, the focus will be limited within the 1.1 Aircraft subsystem. The support. The level 1 to level 2 system hierarchy follows the same pattern as both Aircraft variation subsystems are identical at the level 2 subsystems. This can be seen in the following diagram.

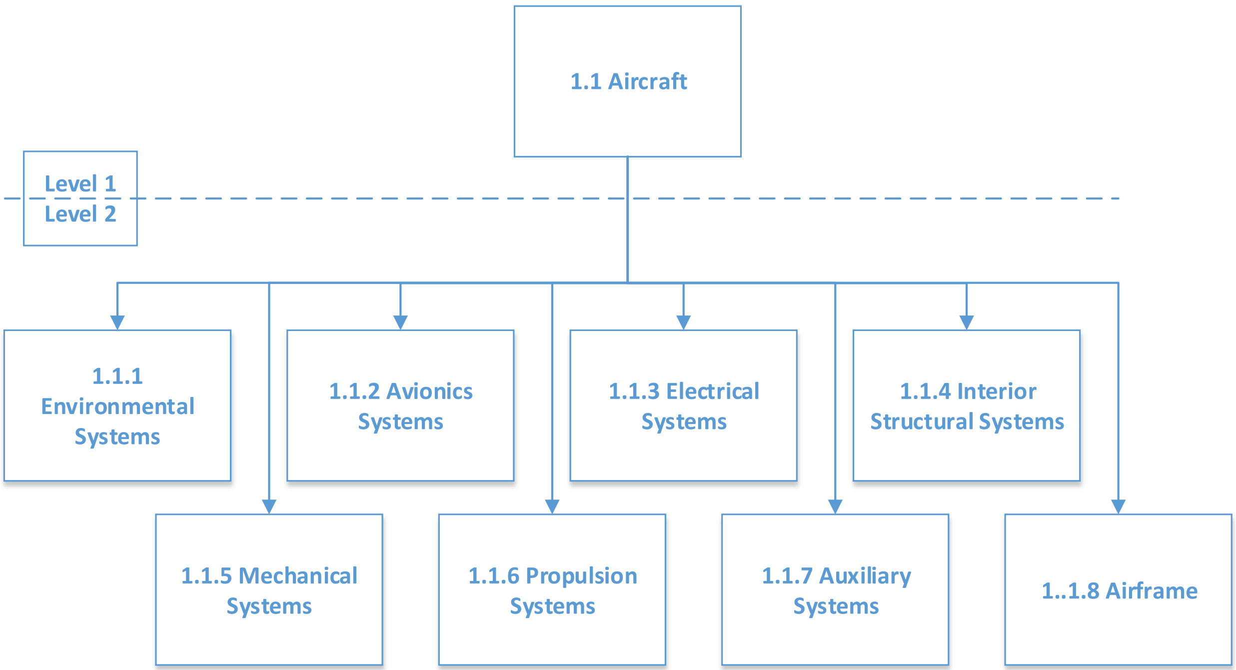

Figure 2. Level 1 to Level 2 System Hierarchy

As one can intuitively infer, the level 2 and level 2 system hierarchies for the commercialized and militarized versions of the 737 are identical. It is at the level 3 subsystems that there is a difference between the aircrafts. This is described is more detail in section 2.

1.2. SOS context diagrams

Now that the level 0 through level 2 systems have been defined the context diagram can be developed. The SOS context diagram can be seen below.

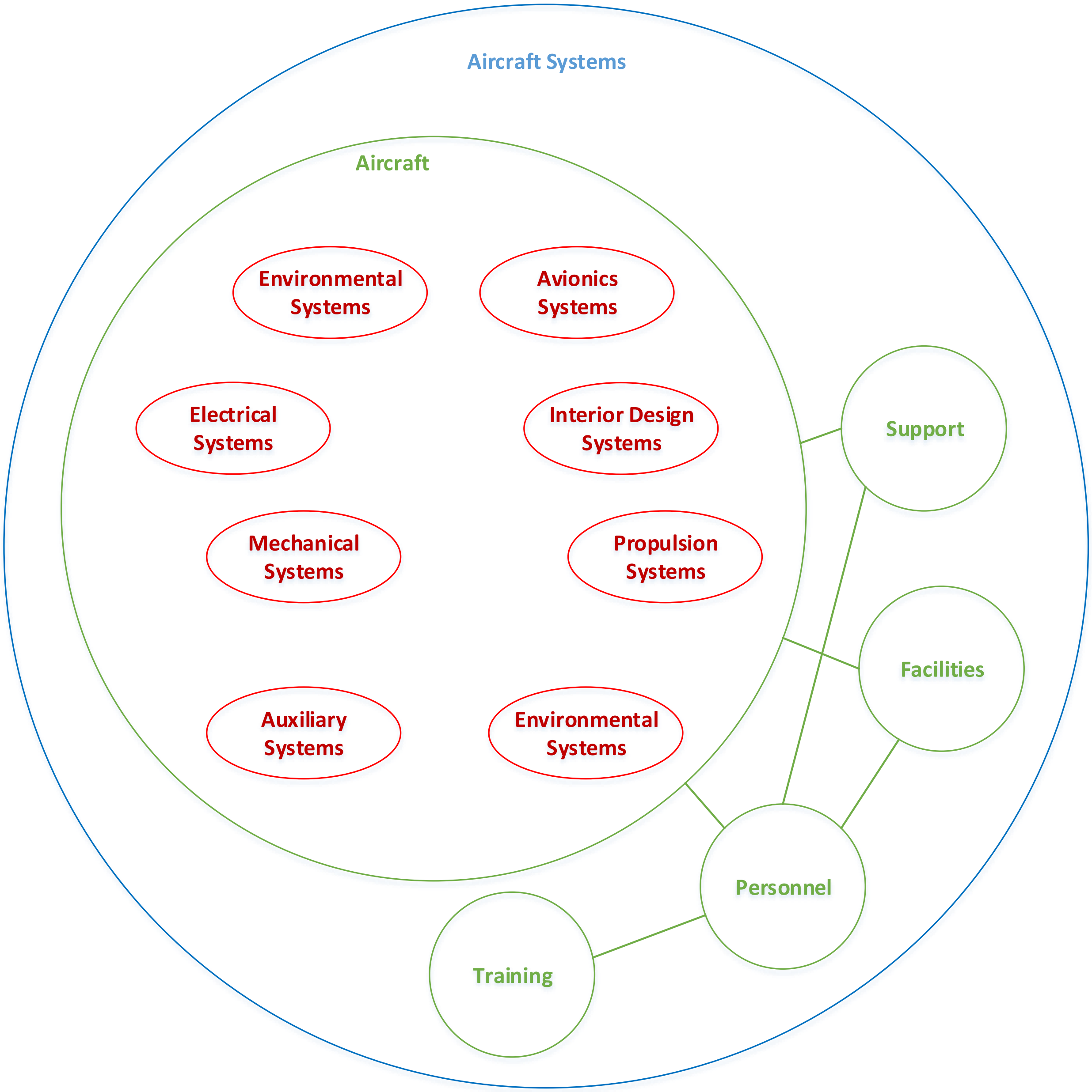

Figure 3. SOS Context Diagram

It is important to note that the size of the subsystems within the context diagram are irrelevant of its physical size or important, the aircraft subsystem is larger because is it our system of interest. The connection between systems represent interfaces between each other the higher level systems.

1.3. Systems Missions and Roles/ConOps

When a system is designed the role of the system is laid out in a concept of operations and thus the system has a specific mission. The 737 was designed to complete a certain mission but when it was modified into its militarized counterpart it possessed different capabilities and thus had different mission roles. A high level description of the concept of operations for both aircraft types is shown in the diagram below.

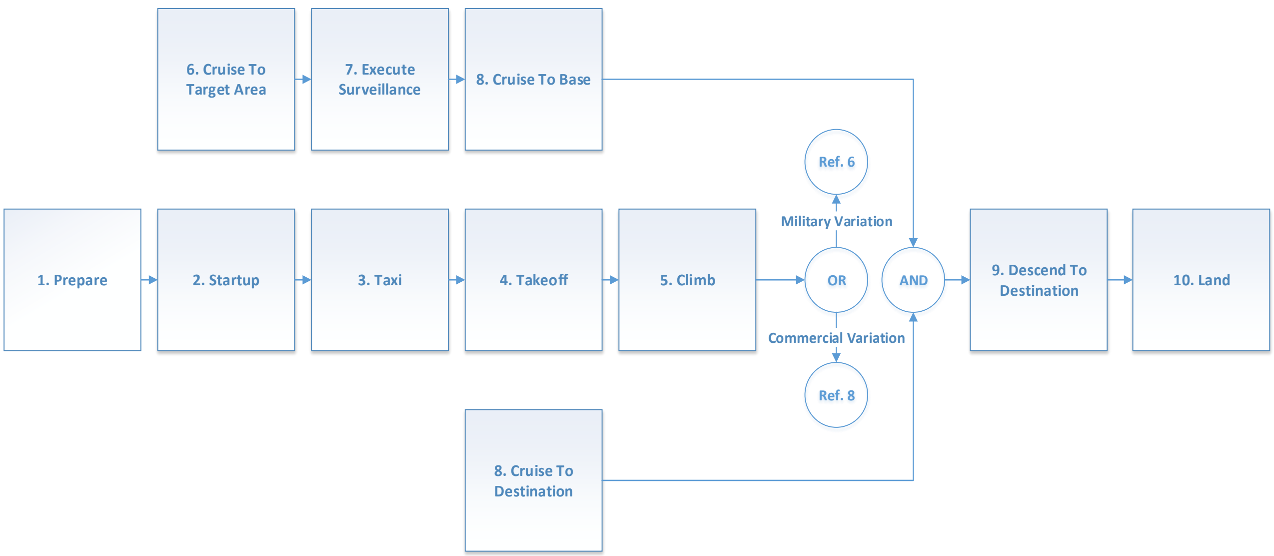

Figure 4. Concept of Operations for both aircraft variations

In order to give an accurate interpretation of the results a specific mission must be defined. Limitations of the mission profile input for the ACS tool and PowerFlow toolset limits the concept of operations. Despite some limitations in the software's ability to create in depth Concept of Operations, the mission profiles were developed to emulate the Concept of Operations as close as possible. The exact inputs to both the ACS software and the PowerFlow toolset for the mission envelope are detailed more in their respective sections.

The mission profiles for the two aircraft variations are kept the same besides the distance of the cruise portion. This is because the militarized variation of the 737 is designed under the requirement of having a longer range.

2. Subsystem Level Overview

2.1. Subsystem Level Comparison

The differences between the aircraft variations becomes evident at the level 3 subsystems. The level 3 system hierarchies are shown in the figures below. It is important to note that the systems that belong to both the military and commercial variations are indicated by solid connectors whereas the systems that belong just to the military variations are designated by dashed connectors. Systems belonging to just the commercial variations are labeled by a double solid connector. This method is used to avoid redundant system hierarchies. It is also important to note that level 3 subsystems such as communications and electronic warfare self-protection have been broken down into their level 4 subsystems because these systems are modeled in the PowerFlow simulation.

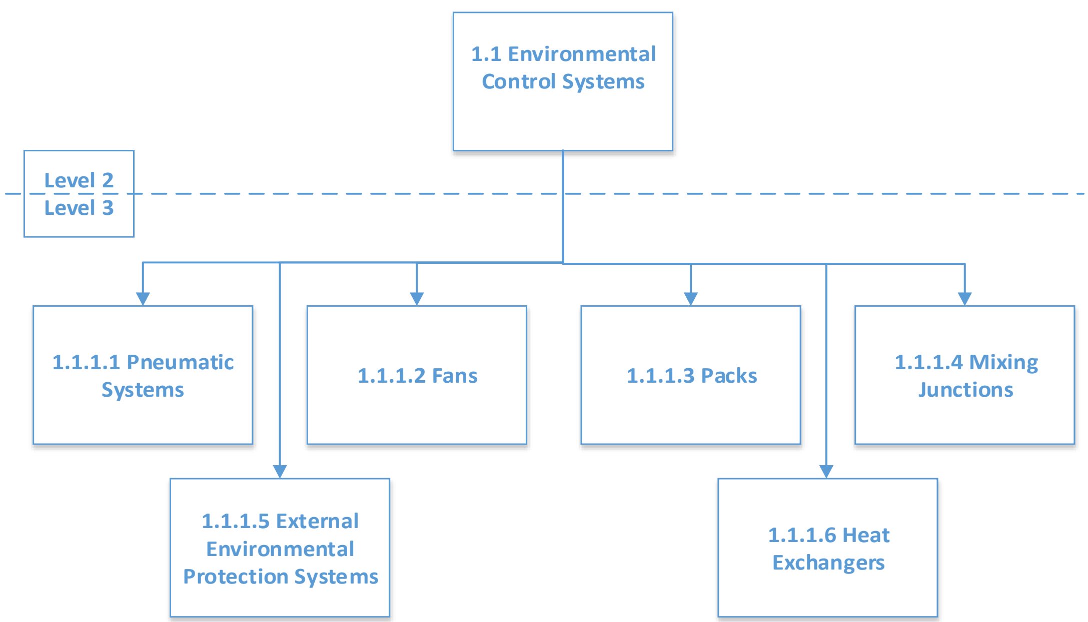

Figure 5. Environmental Control Systems Breakdown

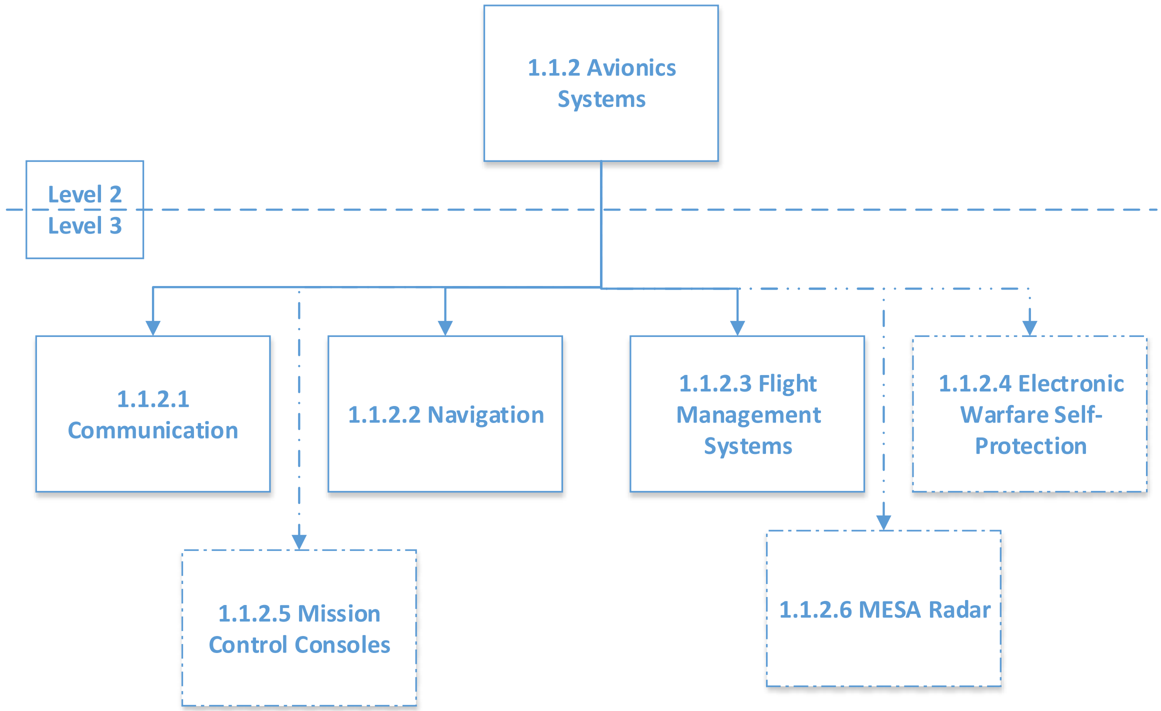

Figure 6. Avionics Systems Breakdown

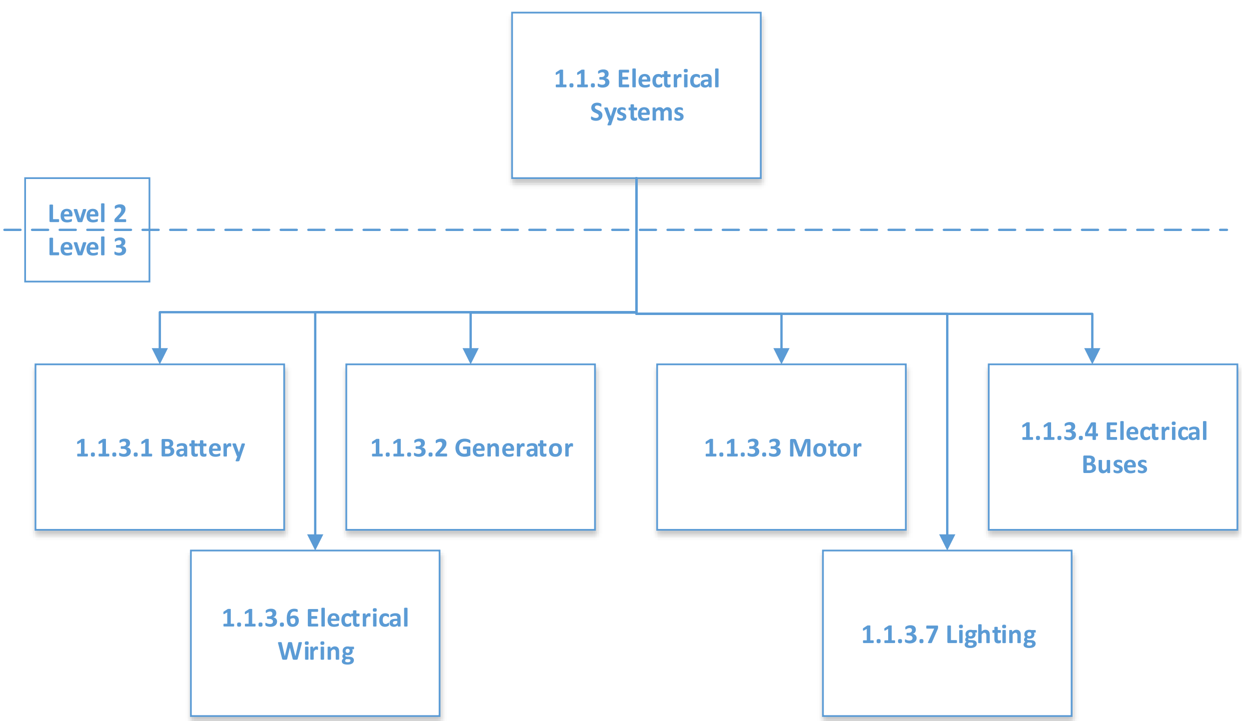

Figure 7. Electrical Systems Breakdown

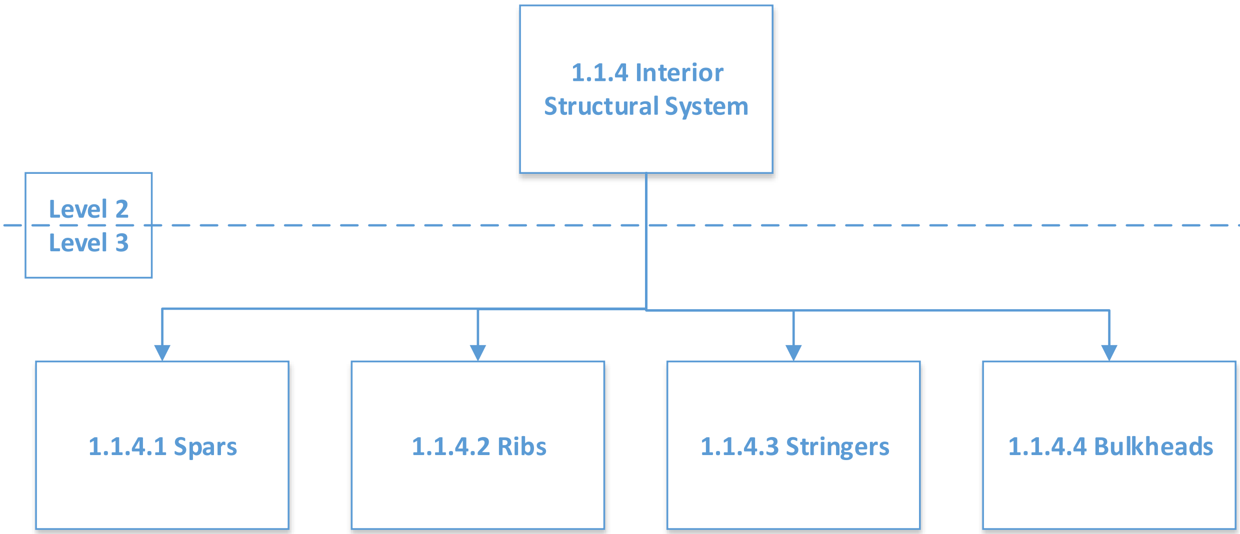

Figure 8. Interior structural systems breakdown

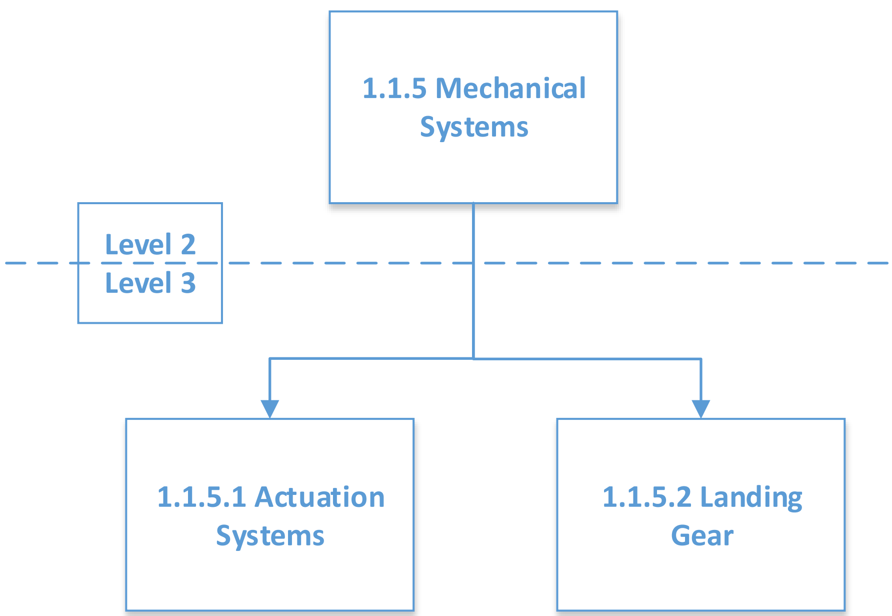

Figure 9. Mechanical Systems Breakdown

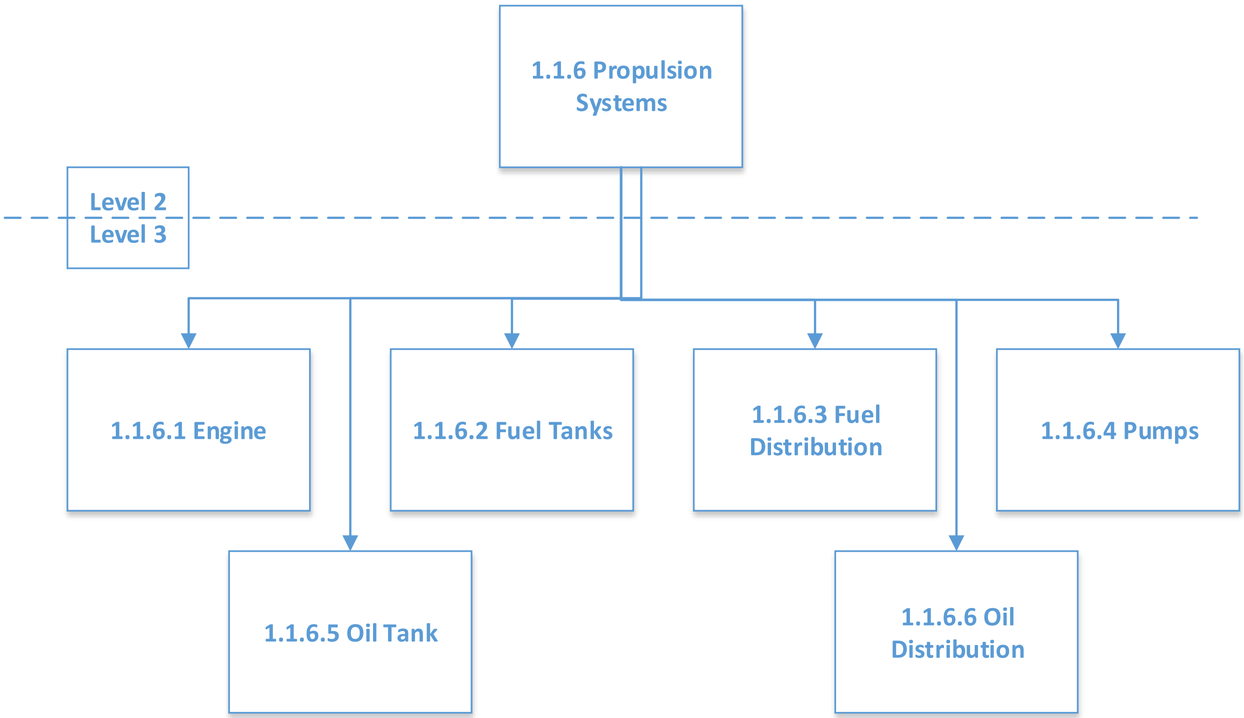

Figure 10. Propulsion Systems Breakdown

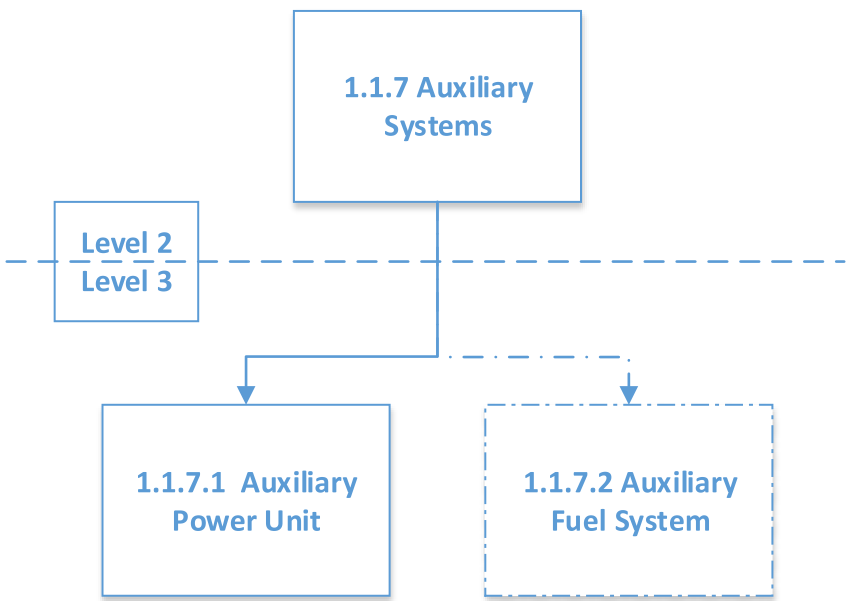

Figure 11. Auxiliary Systems Breakdown

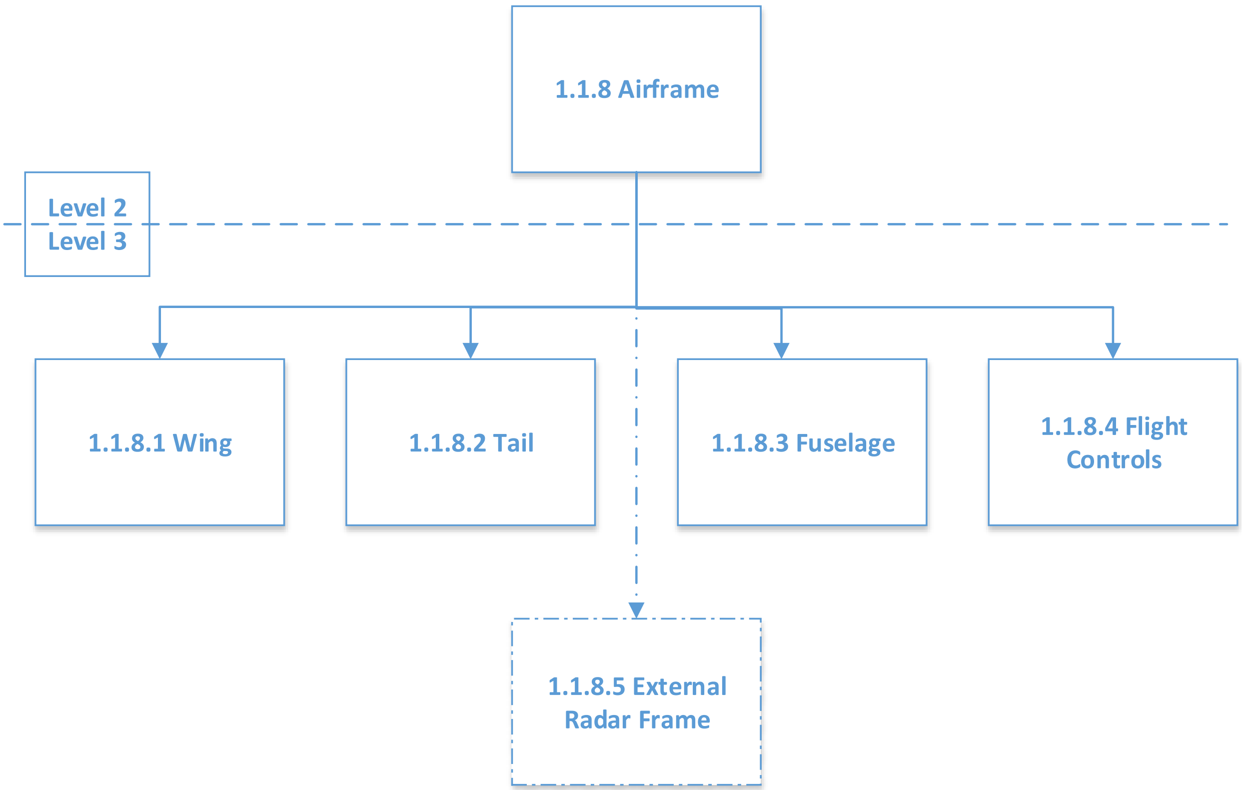

Figure 12. Airframe Systems Breakdown

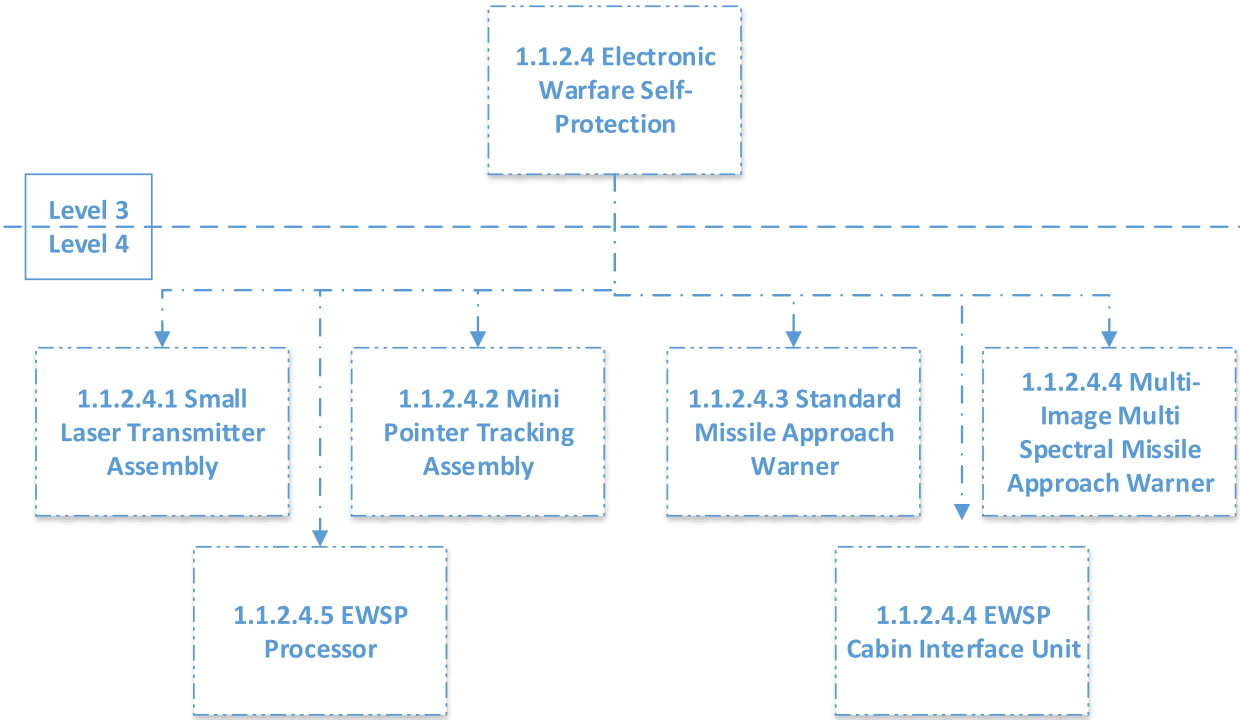

Figure 13. Electronic Warfare Self-Protection System Breakdown

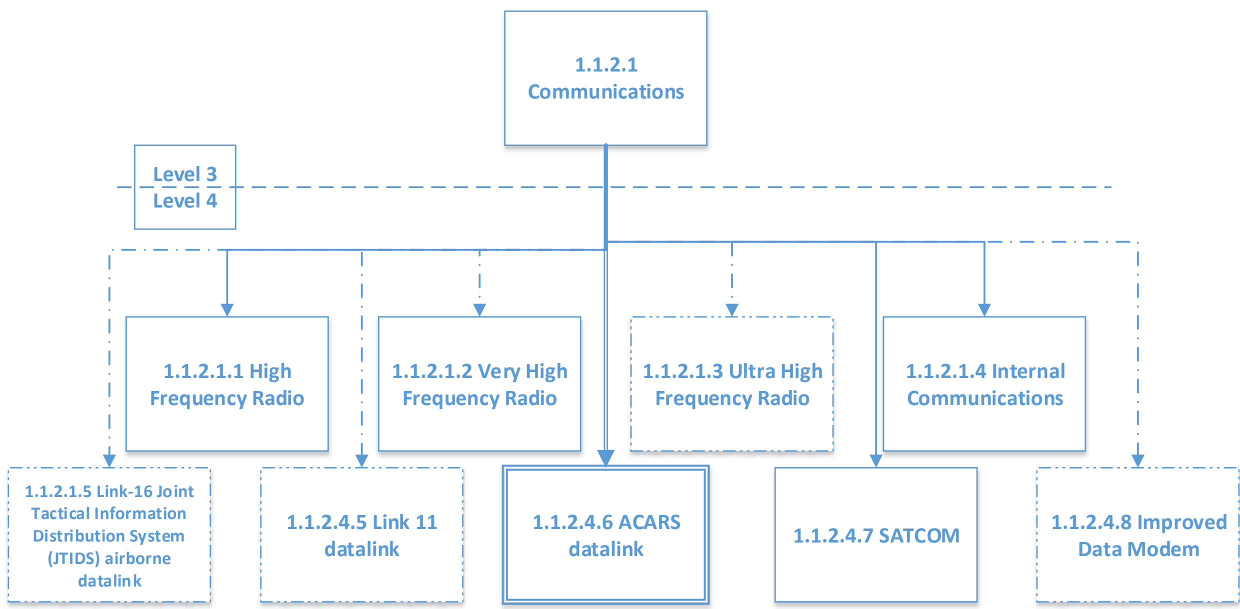

Figure 14. Communications System Breakdown

2.2. Product Interfaces

Now that the subsystems have been defined for both variations it is important to establish which products interface with each other. This is important because the fidelity of the subsequent model used to simulate these subsystems will be compromised if these product interfaces are not reflected within the model itself. Product interface diagrams are used to show either a physical relationship or an exchange between products.

It is important to note that the product interfaces for both the commercial and militarized version are represented together in the product interface diagram below. This product interface diagram is a figure to show the interfaces between the level 3 subsystems. Therefore there are only a few additional subsystems for the military variation: the EWSP systems, mission control consoles, auxiliary fuel tanks, and external radar frame.

		Pnuematic Systems	Fans	Packs	Mixing Junctions	External Environment Control	Heat Exchangers	Communications	Navigation	Flight Management Systems	EWSP Systems	Mission Control Consoles	Battery	Generator	Motor	Electrical Buses	Electrical Wiring	Lighting	Spars	Ribs	Stringers	Bulkheads	Actuation Systems	Landing Gear	Engine	Fuel Tanks	Fuel Distribution	Pumps	Oil Tank	Oil Distribution	Auxiliary Power Unit	Auxiliary Fuel Tank	Wing	Tail	Fuselage	Flight Controls	External Radar Frame
Environmental Control System	Pnuematic Systems		x	x	x		x																														
	Fans			x	x		x																														
	Packs				x		x																														
	Mixing Junctions						x																														
	External Environment Control						x																														
	Heat Exchangers																																				
Avionics Systems	Communications											x					x																				
	Navigation																x																				
	Flight Management Systems										x						x																				
	EWSP Systems											x					x																				
	Mission Control Consoles																x																				
Electrical Systems	Battery															x	x														x						
	Generator															x	x								x												
	Motor															x	x																				
	Electrical Buses																x	x																			
	Electrical Wiring																														x						
	Lighting																																x	x	x		
Interior Structural Systems	Spars																																x	x	x		
	Ribs																																x	x	x		
	Stringers																																		x		
	Bulkheads																																		x		
Mechanical Systems	Actuation Systems																																			x	
	Landing Gear																																x		x		
Propulsion Systems	Engine																										x						x				
	Fuel Tanks																										x	x									
	Fuel Distribution																											x				x					
	Pumps																												x	x		x					
	Oil Tank																													x							
	Oil Distribution																																				
Auxiliary Systems	Auxiliary Power Unit																																				
	Auxiliary Fuel Tank																																				
Airframe	Wing																																		x	x	
	Tail																																		x	x	
	Fuselage																																				x
	Flight Controls																																				
	External Radar Frame																																				

Figure 15. Product Interface for both aircraft variations

2.3. Top Level Requirements

The top level requirements (i.e. the system of system level requirements) for the 737-C and 737-M must be different because the concept of operations for the two systems are different. The following two tables highlight a few example top level requirements that the models must satisfy and will determine the design of the systems at later stages of development. Note these are simply example top level requirements, our analysis will only be able to confirm a select few performance requirements such as range and internal power consumption.

It should be noted that specific parameters are denoted by unknown variables. These variables will be specified later in the design process. However, these models and simulations will assist in determining the design space that still satisfies these requirements.

Table 1. Example top level requirements for the 737-C.

Requirement	Rationale	Type
The 737-C shall be able to control internal environment temperature and pressure.	The 737-C will be designed to transport passengers thus the system must be able to regulate internal environment for comfortable travel.	Functional
The 737-C shall be capable of a ferry range of X nmi.	The 737-C must be able to travel certain lengths to meet customer demands.	Performance
The 737-C shall be capable of anti-icing in a mission operating conditions.	Development of ice can be a large threat to many of the 737-C subsystems.	Regulatory
The 737-C shall emit a maximum of X dBs during all operating conditions.	In commercial aircraft noise can cannot be too high so as to not disturb the external environment.	Regulatory

Table 2. Example top level requirements for the 737-M

Requirement	Rationale	Type
The 737-M shall be able to sustain power consumption at all operational conditions.	The 737-M has different operating conditions with its extensive addition of defense avionics and thus must be capable to supplying power to all avionics in all conditions.	Functional
The 737-M shall be able to target any aerial targets within XX nmi diameter.	The 737-M is mainly used as an early warning aircraft and thus should be able to target and identify all aerial beings within a certain diameter.	Performance
The 737-M shall be capable of a ferry range of Y nmi.	The 737-M must have a ferry range that meets the characteristics of an early warning aircraft.	Performance

As can be seen from the two tables above the 737-M and 737-C top level requirements are dissimilar. This is because the concept of operations of the two systems are different. The 737-C is a transport vehicle designed for commercial use so it makes sense that this small sample of requirements contains a mix of performance, functional, and regulatory requirements. The 737-M is an aerial vehicle designed for early reconnaissance and tracking. As an example, it can be seen that the ferry ranges for the two systems are different. This is because the two systems require different ferry ranges to complete their operations.

3. ACS Modeling

3.1. Introduction to ACS

The Aircraft Synthesis (ACS) toolset is a program is an aircraft synthesis program used to early design phase applications and was developed by AVID LLC [1]. The toolset is based on the NASA ANSYNT toolset and is programed in the FORTRAN language. The input file is a simple text file written in FORTRAN code and is broken down into several modules. The modules include aircraft geometry, trajectory, aerodynamics, propulsion, stability, weights, advanced aero methods, economics, and takeoff. The user is capable of choosing which of the modules will be used in the design of the aircraft. For this study the following modules will used to describe the commercialized 737 and the militarized 737: geometry, trajectory, weights, aerodynamics, and propulsion. ACS outputs a text file that outlines the outputs for each module. The outputs that are of interest within this study are the outputs that were previously listed as the modules that will be used to describe the aircraft models and are analyzed in detail in the next section.

3.2. ACS Modeling Inputs and Outputs

3.2.1. Geometry Input/Output

The geometry input for the ACS tool uses simple inputs to develop individual pieces of the pieces of the aircraft in the form of pods, bays, horizontal tails, vertical tails, and canards. The only difference between the 737-m and 737-c is the 737-m possesses the MESA radar above the aft end of the fuselage. Unfortunately, this was unable to be modeled within the geometry input. In order to compensate the radar was modeled within PAGES. PAGES is a separate tool developed by AVID LLC that developed 3-D models of the early aircraft designs. It can take in the geometry output from the ACS tool and develop a representative visual model. The MESA radar was added within the PAGES model on top of the already existing geometry that was imported from the ACS output. The additional drag from the radar was compensated for in the aerodynamics module and will be discussed in more detail in the Aerodynamics Input section.

737 geometry values were all obtained from the 737 Technical Guide [2]. The specific variables used in the input were as follows:

Table 3. Geometry Input for ACS wing models

Variable	Wing	Vertical Tail	Horizontal Tail
Sweep (deg)	25.0	30	30.0
Taper Ratio	0.159	0.271	0.203
T-C ratio at root	0.120	0.100	0.100
T-C ratio at tip	0.120	0.100	0.100
Dihedral Angle (deg)	6.00	N/A	N/A
Area (ft^2)	1340	284	352
AR	9.45	1.91	6.16

Table 4. Geometry Input for ACS fuselage

Variable	Fuselage
Diameter of Fuselage (ft)	106.0
Length of Fuselage (ft)	12.3
Fineness ratio - afterbody	3.50

The output geometries from the ACS tool are as shown:

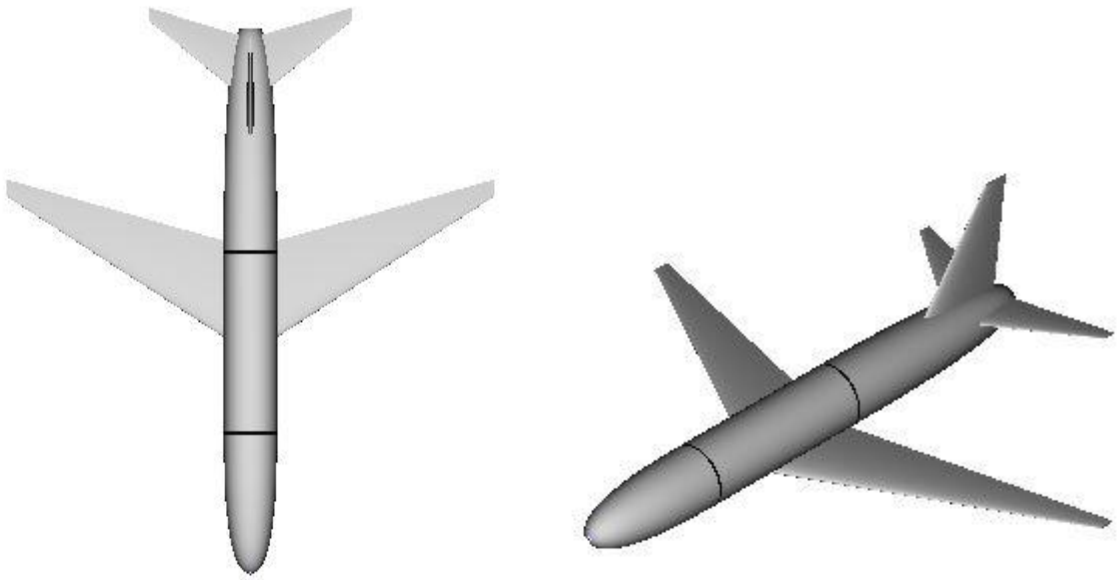

Figure 16. ACS approximation of commercialized 737

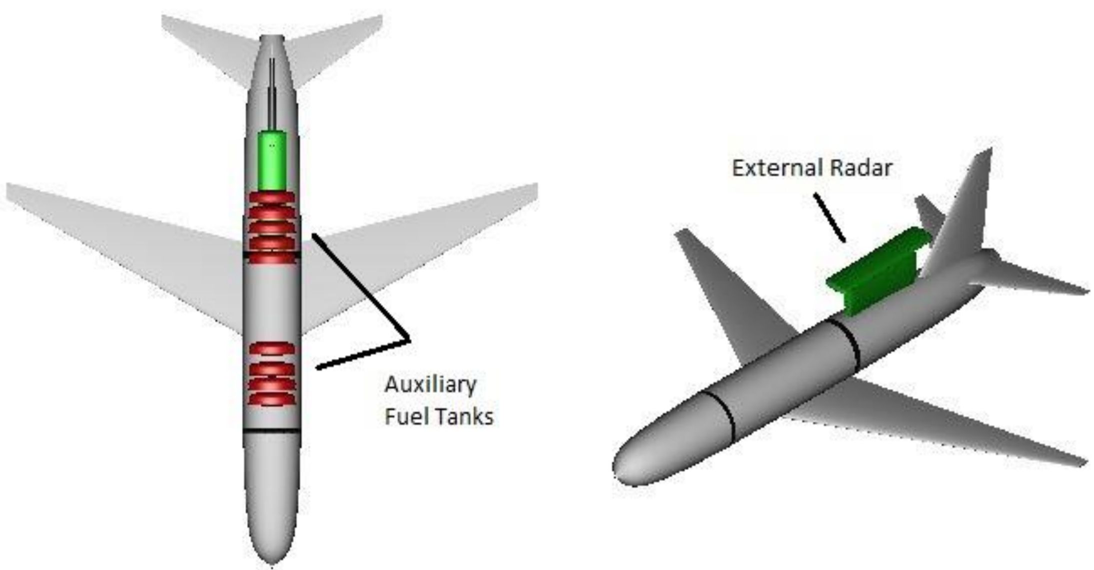

Figure 17. ACS approximation of militarized 737

As discussed previously two prominent systems that exist in the militarized 737 are the MESA radar and the auxiliary fuel tanks. Both of these systems were added to the models after its input into the PAGES toolset. It can be seen from the figure above that the MESA radar is labeled in the full view of the 737-M. The auxiliary fuel tanks and their respective locations are labeled in the top view of the 737-M. It was assumed that the 737-m used the maximum amount auxiliary fuel tanks at 9. From the Boeing technical guide the configuration for a 9 auxiliary fuel tank system can be seen in the top view of the figure above. Note that, although shown in the model, the auxiliary fuel tanks are internal to the airframe.

3.2.2. Trajectory Input

The trajectory input within the ACS specifies the mission envelope the aircraft will be undergoing in order to determine the performance of the aircraft. This module is split into two parts: a namelist format that accepts constants and a formatted input that defines the mission envelope. The trajectory module is capable of calculating the following mission phases: climb, acceleration, cruise, loiter, descent, and hover. Startup to takeoff, however, is packaged as a constant in the in the namelist format section. The mission envelope that was used is as follows:

Table 5. Trajectory Input for ACS models

Phase	Start Mach	End Mach	Start Altitude (ft)	End Altitude (ft)	Distance (nmi)	Time (min)
Climb	.20	.20	0	1500	0	0
Climb	.20	.77	1500	36700	0	0
Cruise	.77	.77	36700	36700	2600 (commercial) 3500 (militarized)	0
Descent	.77	.40	36700	1500	0	0
Loiter	.40	.40	1500	1500	0	20

It must be noted that the table above is simply the input to the trajectory module and does not reflect the actual conditions perfectly due to limitations within the ACS software. It can be seen that the Climb is broken up into two separate legs. This was done in order to represent startup to takeoff. As stated before, startup to takeoff, are bundled into a single constant that is just a time variable to describe the time from startup to end of takeoff.

It can also be seen that the distance and time are zero for climb and descent. For climb, this is the case, because ACS approximates climb using the max R/C assumed for the aircraft. Descent is extremely limited within ACS as it assumes an immediate descent to the given end altitude.

Cruise and loiter lengths are described differently because the ACS input can only accept time as an input for loiter, combat, and hover phases while only accepting distance for cruise. The only difference while developing the mission profiles between the 737-c and 737-m is the cruise distance. This was to give an accurate representation for a ferry mission for each aircraft variation. To see the exact trajectory input, the input file can be found within the appendix I.

3.2.3. Aerodynamics Input/Output

The aerodynamics module is used to calculate the aerodynamic characteristics of the aircraft model. The parameters that describe the main aerodynamic characteristics of the wing-body for the commercial 737 were provided in a sample input provided by AVID LLC. This gave a rough estimate as to what the 737-m should look like in terms of aerodynamics but does not take into account the aerodynamic impact the external radar would have on the system. Within the ACS users guide, it can be found that the determination of drag is obtained by the component build-up method. Given this, it was possible to keep the original aerodynamics supplied with the input file from AVID LLD but build a separate model independently external to ACS to find the drag for the 737-m variation.

Given the known equation for Reynolds numbers:

$$R_e = \frac{V * L}{\nu}$$

From [4] and [3] respectively we get the equations:

$$C_f = \frac{.455}{\log(R_e)^{2.58}}$$

$$C_d = \frac{K * S_{wet} * C_f}{S_{ref}}$$

K is the body form constant and can be estimated from [3]. The kinematic viscosity, ν, is estimated from the U.S. standard atmospheric air properties tables at 35,000 ft. C_f is from a numerical fit called the Karman-Schoenherr Eauation [4]. L can be estimated using the length of the perpendicular face of the body relative to the flow. The length of the MESA radar, which is the main contributor to aerodynamic differences between aircraft variations, is estimated by simply using its height and is assumed to be a pylon. Using these equations we are able to find the parasite drag for each component to build up the total drag for the system.

Note that this consists of only parasite drag and no other drag contributions such as induced drag. This was done to accurately emulate the model used within the ACS tool. Now that it has been assumed that the additional systems for the 737-m have an impact on the aerodynamic characteristics ACS will allow us to analyze these differences.

Based on the method used in reference 13, a MATLAB code was written to change the Mach number within the FORTRAN input text file, run the ACS program, read the output and repeat through an iterative process. The diagram below gives a visual summary of the process executed by the subroutine. For the exact code used to obtain the data please refer to the appendix II.

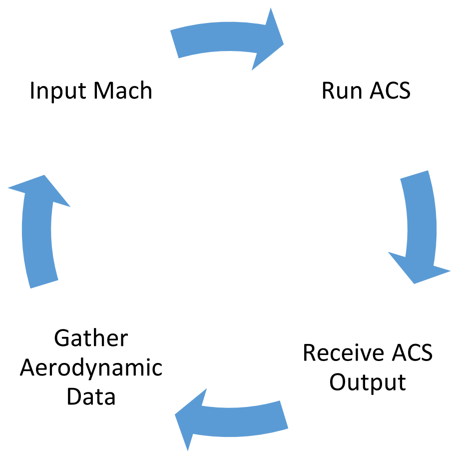

Figure 18. MATLAB subroutine process to gather Aerodynamic Data

From this subroutine it was possible to pull major aerodynamic characteristics that the ACS aerodynamic output produces which are C_L, C_D, and $\frac{L}{D}$. Pulling the aforementioned aerodynamic characteristics from the ACS output file it was possible to obtain the following figures:

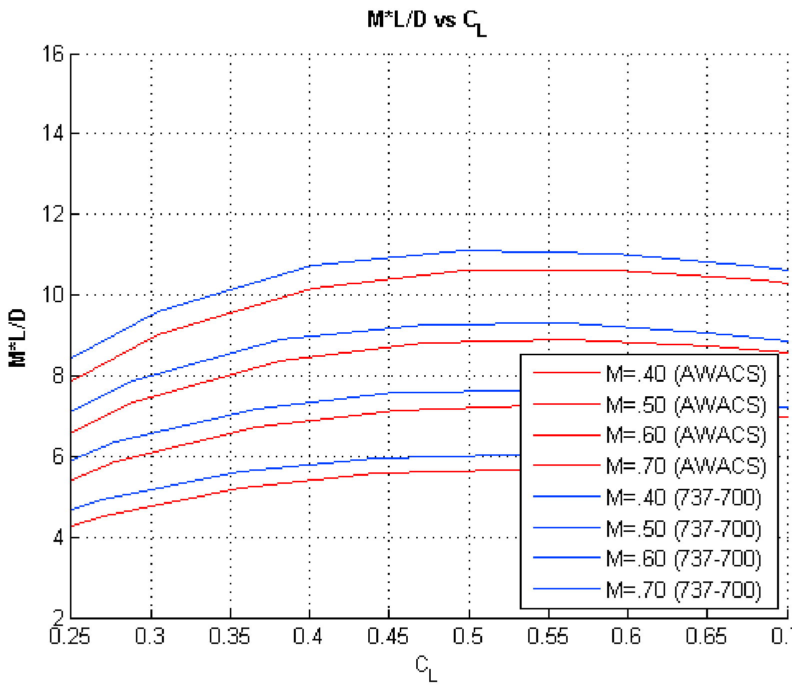

Figure 19. M*L/D vs C_L

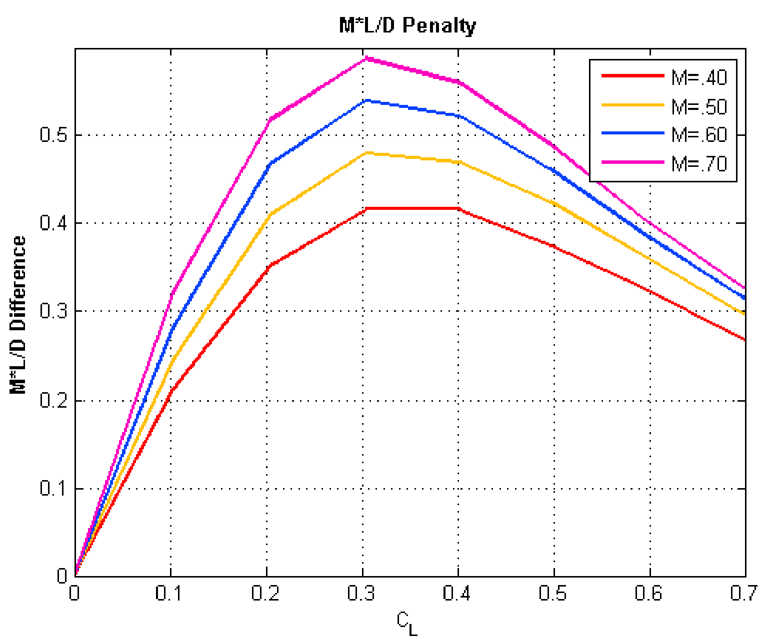

Figure 20. M*L/D Difference between systems

Figure 20 offers a visual of the penalty the geometrical configuration of the 737-M has on the aerodynamic performance. This analysis is important as it highlights better operating conditions for the 737-M. As stated earlier in the study the 737-M was designed from the 737-C structure so there are many aspects of the system that cannot be changed. Therefore, it is important for designers to understand, under what flight conditions, does this modified system incur the greatest penalty in performance.

It can be seen from figure 20 that the 737-M incurs a greater penalty when mach number increases. It can also be seen that the penalty increases to a maximum at a C_L of .3 and then again begins to decrease. Thus it is safe to say that a 737-M would operate best in conditions where the lift coefficient is furthest from .3.

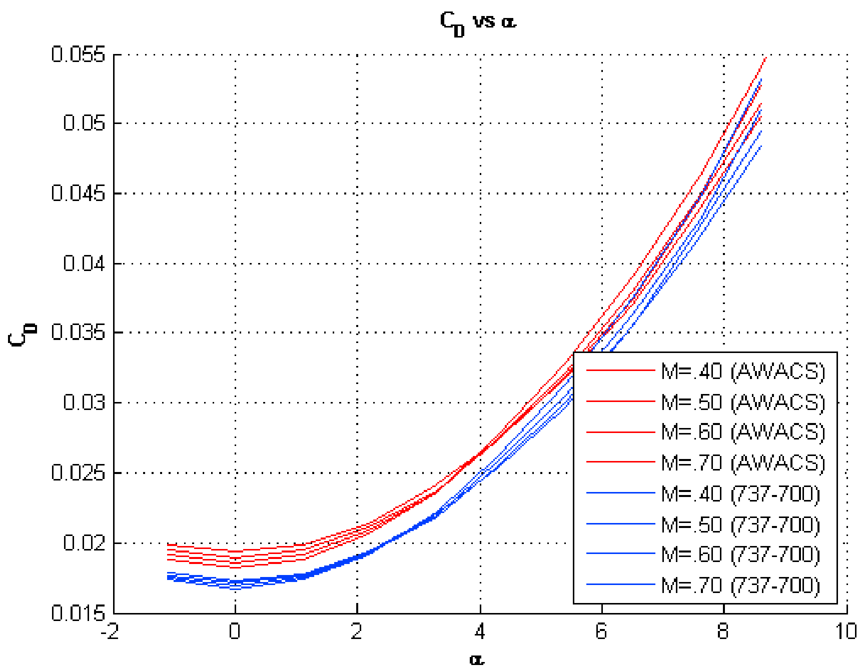

Figure 21. C_D vs α

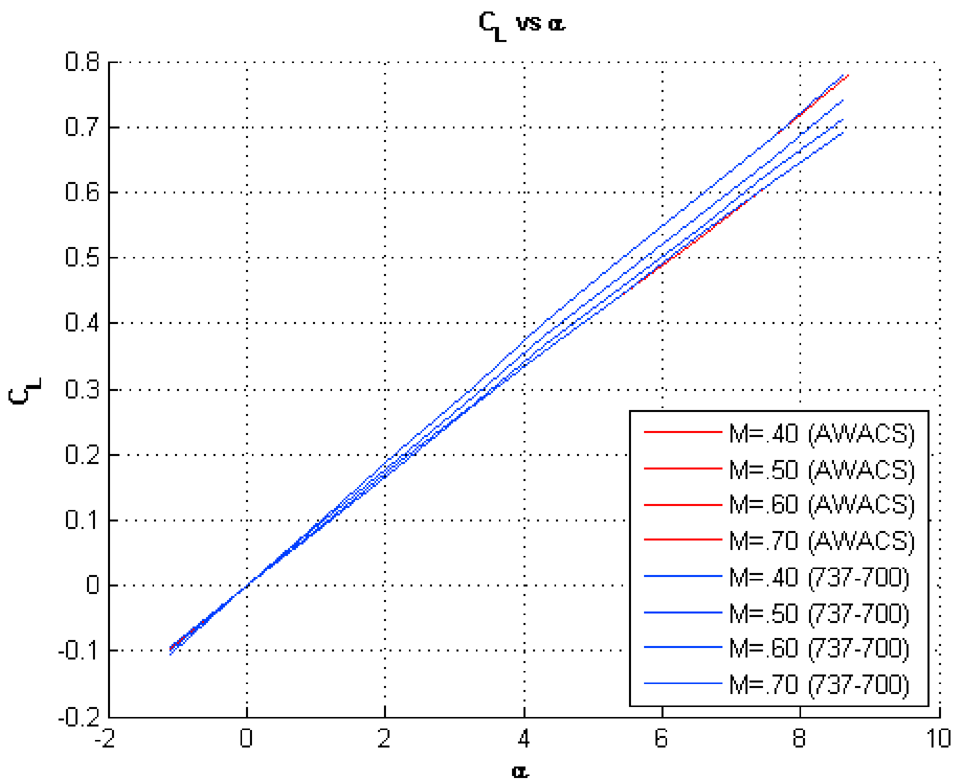

Figure 22. CL vs α

It is important to note that all of these calculations were done at an altitude of 36,700 feet which is the altitude of cruise used for both mission envelopes. It can be seen from the figure 19 above that value $\frac{M*L}{D}$ is always larger for the 737-c regardless of the mach number or of the lift coefficient. This could be due to either the Lift for the 737-C being larger or the drag for the 737-M being larger. Through deductive reasoning we can deduce that this is because the drag for the 737-M is larger. This is what was to be expected due to the additional parasite drag induced from the external radar system. Looking at figure 21 it can be seen that the drag coefficient for the 737-M is always larger than the drag coefficient for the 737-C.

The geometries for the aircraft models are nearly identical therefore it should be insinuated that the lift coefficients would be the same. Looking at figure 22, it can be seen that this holds true for the vast majority. However, when alpha is very high at around 6 to 8 degrees, the lift coefficients for the 737-M drop slightly, but visibly lower, than the 737-C. It is known that the lift coefficient is related to air density, air density, and the wing geometry. In this experiment, all of those variables were the same for each model. The variance may be due to a more complex algorithm used by AVID LLC underneath the hood of the ACS software. Regardless of this slight variation the lift coefficients were nearly exact as we would expect.

This slight drop in the lift coefficient at high angle of attacks could have an impact on the performance of the 737-M during the climb phase. However, it can be seen the penalty difference between the 737-C and 737-M is lower than other angle of attacks in figure 24. These aspects are something that should be taken into account when designating the climb phase of a 737-M mission profile.

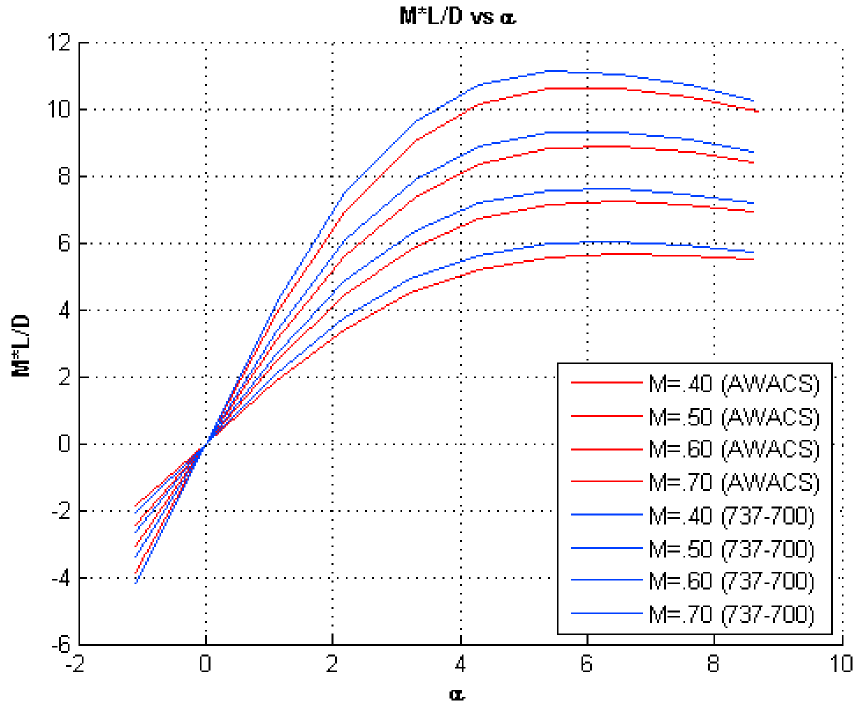

Figure 23. M*L/D vs α

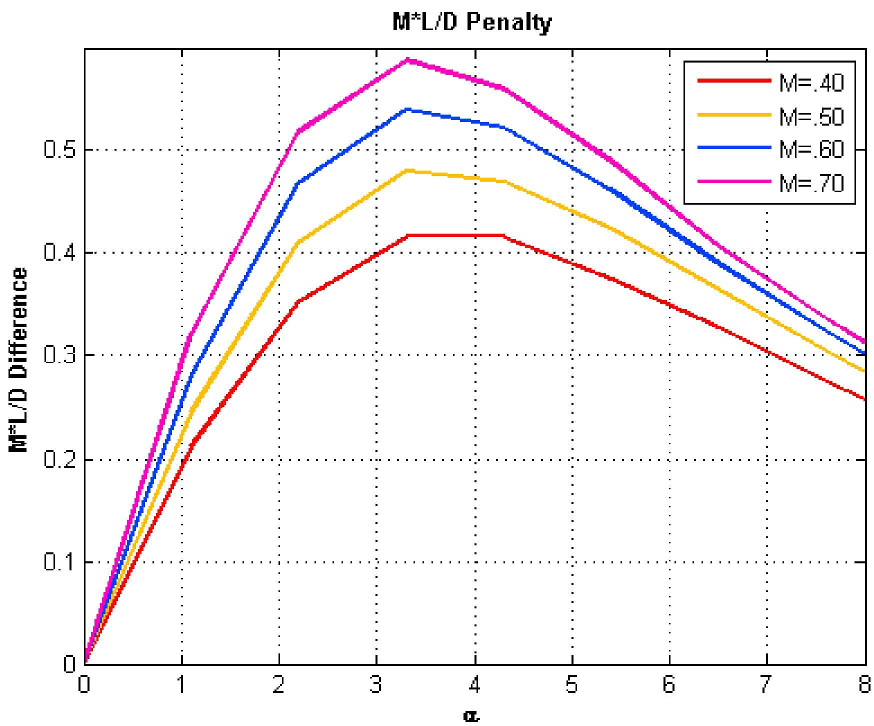

Figure 24. Angle of attack penalty

It can be seen that the penalties from the angle of attack and the lift coefficient are quite similar. This is because the lift coefficient is dependent upon the angle of attack and thus yield similar results. Just like the lift coefficient the penalty behaves in an almost parabolic behavior; peaking just over 3 degrees then dropping thereafter.

For the aforementioned reasons, it was possible to validate that the drag for the 737-m would increase due to changes in its subsystems. By developing an external and independent model for developing parasite drag it was possible to not only validate that the 737-m induced more drag but verify that the ACS software meets the specifications of analyzing aircraft aerodynamics.

3.2.4. Weights Input

The weights input was a fairly simple input for the ACS tool. ACS is capable of taking in different values of fixed weights. However, since this study is not built around designing a new aircraft so the initial take off gross weight guess can be assigned. ACS finds the aircraft weight through iterating several of the modules to converge on a single solution with the takeoff gross weight estimate as the initial guess. The takeoff gross weights were found from the 737 technical guide and used as an initial guess. After a few iterations the ACS module found the takeoff gross weights to be the following:

Table 6. Weights Estimation from ACS

Aircraft	Initial Guess Weight	Final Convergence Weight	Percentage Error
737-m	171,000 lbs	170453.1 lbs	.32 %
737-c	154,500 lbs	156011.1 lbs	.92%

Given the low percentage errors between the final convergence weight of the ACS model build and the actual weight of the aircrafts, this solution was used. Errors within the weights could come from smaller systems that are not built into the ACS functionality. ACS only accepts a range of system weight inputs thus cannot possibly take into account all weighting factors.

It can be seen that the initial and final weights are fairly similar and we can assume that the weights within the ACS tool converged accurately and thus are geometrical models are correct. There were also fuel weights that the ACS input processed. From the 737 technical guide it was found that the 737-c has a fuel weight of 45856.2 lbs. For the 737-m the amount of fuel was estimated. From a Boeing media release statement if can be read that the auxiliary fuel tank systems that can be equipped on Boeing Business Jets (BBJ) can accompany nine tanks that can hold 520 gallons each accumulating to a total of 3,800 gallons of additional fuel [5]. In order to estimate the fuel the 737-m can hold a few assumptions must be made. We must assume that the 737-m utilized the 9 auxiliary tank configuration and also assume that it holds the same amount of fuel as the 737-c without the auxiliary fuel tanks. Given the aforementioned assumptions, it can be calculated that these additional fuel tanks hold approximately 25460.0

lbs. of fuel. This gives an estimate of roughly 71316.2 lbs. of total fuel for the 737-m. This was then added as a constraint for the weights in the final vehicle convergence.

3.2.5. Propulsion Input/Output

The propulsion inputs for the ACS toolset is used to design and compute the performance of the aircraft engine. The propulsion input takes two primary inputs: engine type and computation format. Computational format determines whether the tool uses cycle analysis or previously determined LEWIS table lookup files. For this study a cycle analysis is used for both aircraft models. For the engine type a Generic Bypass Turbofan 9300 lb. class was selected as to be the closest fit to the CFM56 turbofans found on the actual aircrafts.

The propulsion module also takes in geometric and propulsion related parameters. The relevant parameters were found from references 6 and 7 and are listed in the table below.

Table 7. Propulsion Parameters for ACS

Aircraft	Engine Type	Thrust	Diameter	Length	Bypass Ratio	Overall Compressor Ratio	Power Extraction
737-m	CFM56-7B27A	27,300 lbs.	5.08 ft.	8.23 ft.	5.1	28.9	Varied
737-c	CFM56-3B2	20,600 lbs.	5.08 ft.	8.23 ft.	5.5	22.7	Varied

As can be seen from the table above the power extraction for both engine models falls under varied. This is because the power needed for the subsystems of an aircraft is not constant even though the ACS model takes it in as a single constant. The impact of this parameter on the internal and external performance of these aircrafts will be a large part of this study and will be examined in the upcoming sections.

3.3. Aux Results

The ACS propulsion module accept the power extracted from the inner turbine. Since the subsystems differences within the 737-m mainly consists of avionics that will draw electrical loads from generators, it is important to see what the impact of the power extraction has on the performance of the aircraft. To give an initial interpretation, a subroutine was written to visualize the impact of the engine power extraction has on the amount of fuel used. The subroutine functions nearly the same as the aerodynamics code in that it changes the “AuxInnerPower” parameter within the input text file, runs the ACS tool, reads the fuel used, and plots the corresponding data for each aircraft model. The tool was run over a trajectory identical to the mission envelope laid out in the trajectory input/output section. The exact code can be found in Appendix V. The results were as follows:

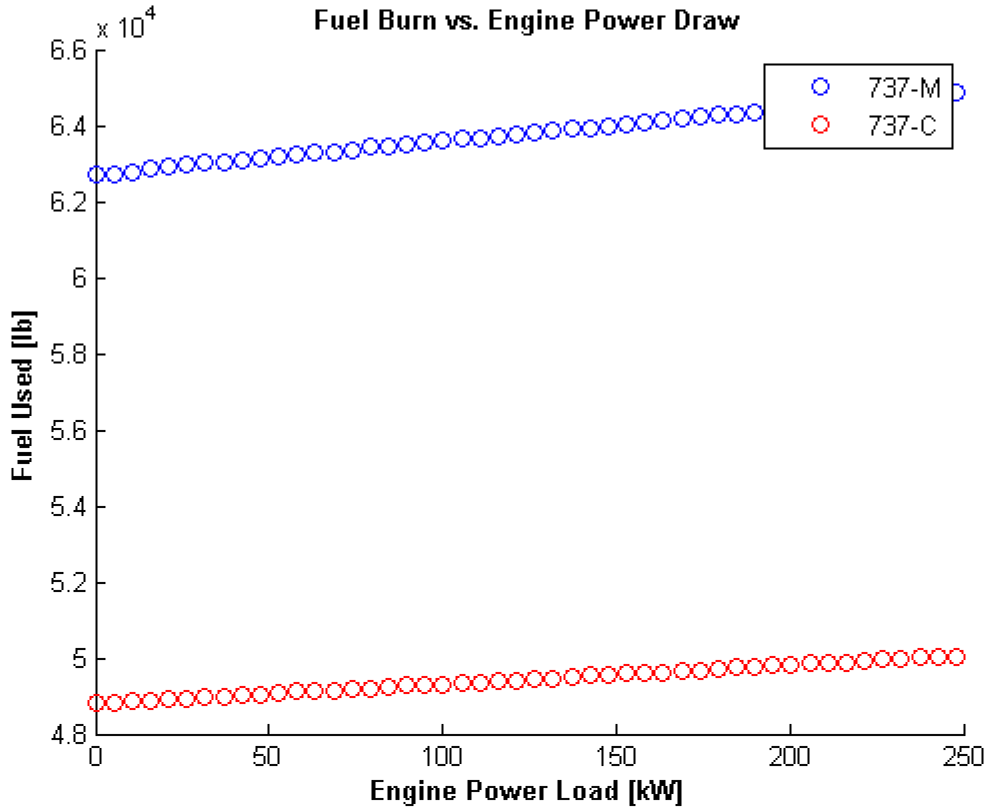

Figure 25. Engine Power Extraction vs. Fuel Used for Both Variations

Table 8. Fuel Weight and Aux Analysis

Aircraft	Fuel at 0 kW extraction	Fuel at 250 kw extraction	Fuel Difference	Percent Difference	Rate [lb/kW]
737-m	62688 lbs.	64869 lbs.	2181 lbs.	3.48 %	8.72
737-c	48806 lbs.	50051lbs.	1245 lbs.	2.55 %	4.98

It can be seen from the table above that the power extraction does make a difference. For a ferry mission for the 737-c the difference between a constant 0 kW engine extraction and a 250 kW engine extraction creates a 2.5% and greater than 1200 lb. of fuel differential. This number is even higher for the 737-m since its ferry mission is higher and an overall higher amount of fuel is used.

An important output is the rate at which fuel burn is used as engine draw increases. The fuel rate for the 737-M is 3.74 $\frac{lb}{kW}$ higher than that of the 737-C. This indicates that the penalty for an increased power draw from the engine is greater for the 737-M. This conclusion emphasizes that a 737-M will save more fuel if the engine power load can be lowered throughout the mission profile.

When looking at the figure above there are a few aspects that are obvious and what were to be expected but there are others that are not so intuitive. One would expect that the higher the engine power load the higher the fuel used. This trend holds true for the vast majority, however, there are instances among both of the aircraft models where an increase in the electrical load results in a decrease of fuel used for the mission. This infers either stochastic variances based around the inputted engine power load or some more complex algorithm used within the ACS software. This may have been implemented into the ACS software in order to simulate mission conditions that could affect the amount of fuel used. In either case, the trend depicts an increase in fuel with an increase engine turbine extraction but as can be seen in the simulation this exists for only the overall trend and not slight difference in the engine turbine extraction.

It can also be seen from the figures above that the variance for the 737-m the variance for the amount of fuel used is larger. This may be due to the higher pound of fuel burned per kilowatt for the. The higher the rate the larger the variance. Therefore, it could be assume that the 737-m has a higher chance of unexpected high amounts of fuel usage.

From the ACS software, it is now understood that an increase of power extraction from the turbine increases the fuel used in general. But that doesn't really give an idea of how that affects the mission of the aircraft models. To illustrate this idea the following figures were calculated:

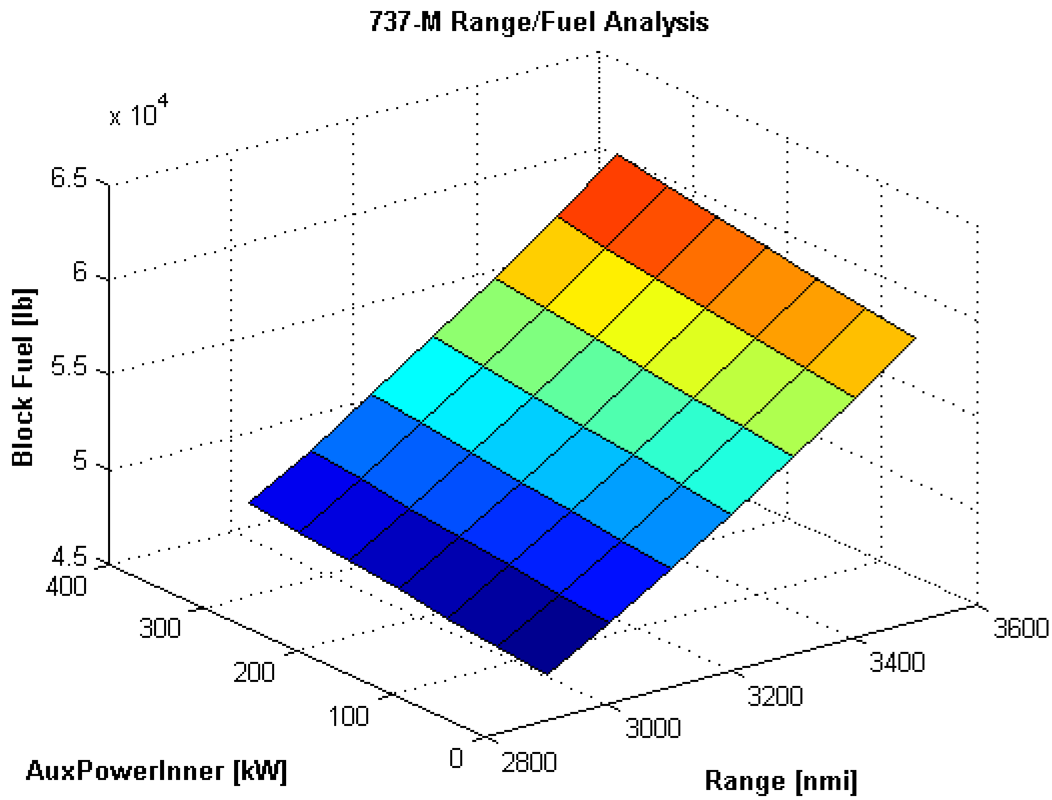

Figure 26. 737-M Range and Fuel Surface Plot

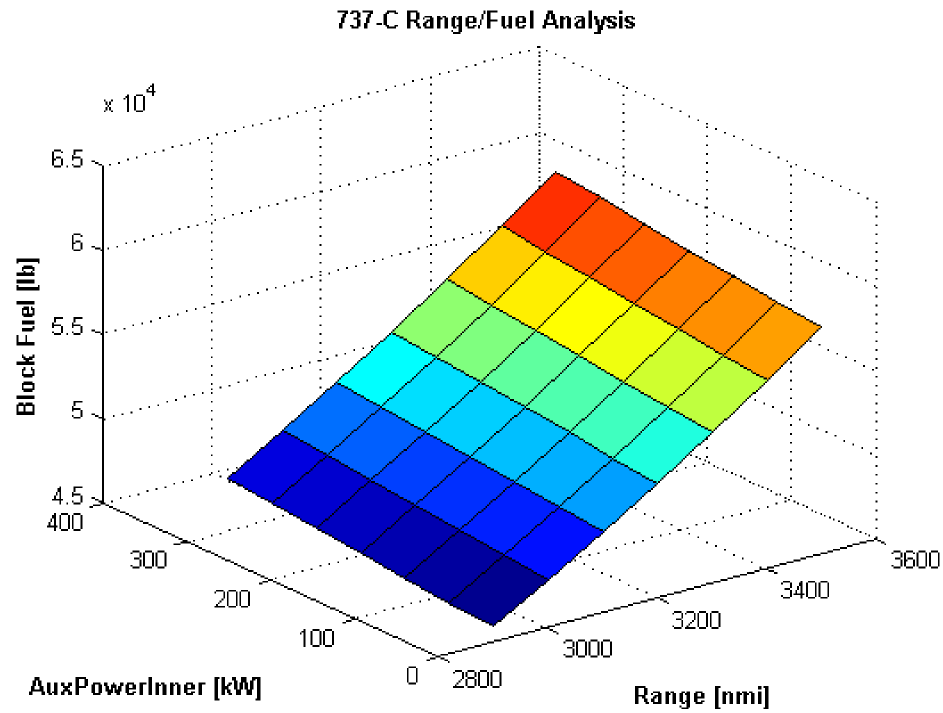

Figure 27. 737-C Fuel and Range Surface Plot

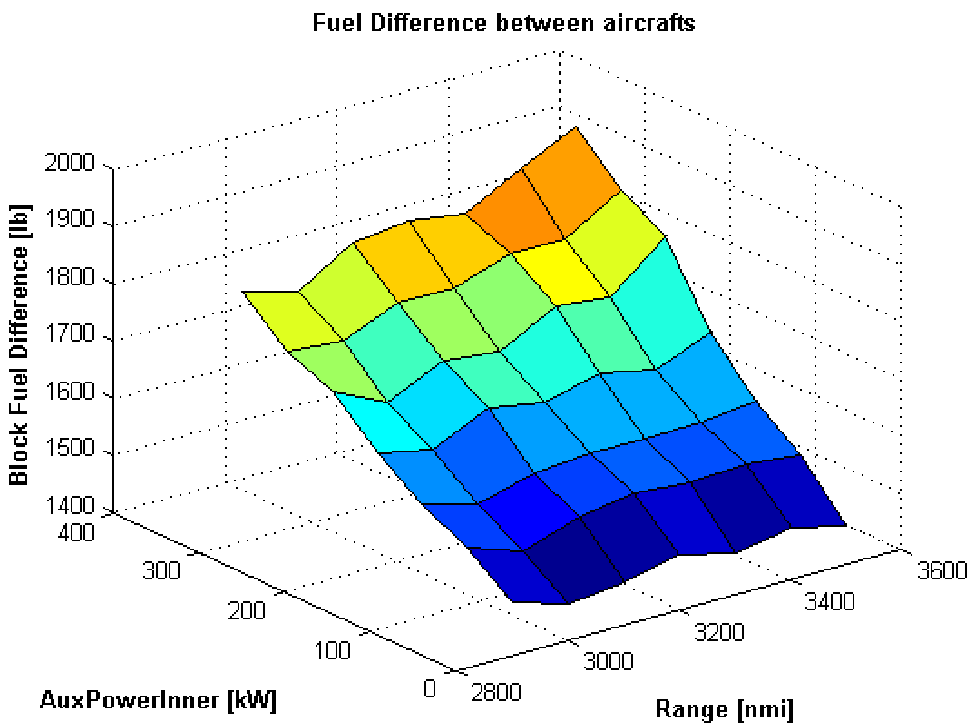

Figure 28. Fuel Differential between aircraft variations.

The figures above the developed using a code similar to the one used to determine the effect of power turbine draw for the aircrafts and can be found in Appendix IV. An additional loop was developed in order to ascertain the effects of both the power turbine draw and range. In the previous section, it was determined, form the ACS toolset, that the amount of fuel needed for a mission increases with an increasing power draw from the engines turbine. This can also be seen by the variation of colors on the surface plots as the power draw increases. While the power draw does increase the amount of fuel needed, it is far less of a factor on overall fuel than range is. This can be seen by the slope of the range vs. fuel axis is much higher than the power draw vs fuel axis slope.

It may be difficult to see differences between the amount of fuel used between aircraft variations in figures 27 and 28. However, it can be seen that the surface is raised higher than that of its commercial counterpart. This means that based solely on the inputs of the ACS toolset the militarized variation of the 737 requires more fuel than its commercial counterpart even when parameters such as range and the power draw from the turbine are the same.

In order to visualize this difference a surface plot was developed in figure 29. It can be seen that an increasing range does not relate to an increasing difference of fuel among the two aircrafts. While fuel differences vary, most likely due stochastic discrepancies built in within the ACS code, it does not increase overall. This results is actually counter intuitive to the anticipated results. The most prevalent reason for these results could be due to the similar geometries of the two aircrafts. Since the aircrafts are being run throughout the same mission envelope with similar geometries this could have some effect but does not explain effects due to aerodynamics characteristics. It is important to note that the external radar for the 737-M was not built into the geometrical dimensions of the model until after simulation due to limitations of the ACS toolset. Its effects on the aerodynamics were modeled independently in the aerodynamics module separate from the geometry but, as can be seen from figure 8 in the previous section, the drag for the 737-M variation is greater than the 737-C. When this is the case the engines from the 737-M would require more thrust to counteract the higher amount of drag induced upon the aircraft. Theoretically, this drag, over time, should increase the amount of fuel the 737-M needs with an increased range. However, this is not observed here.

On the other hand, as the power draw from the turbine increases the difference of fuel used for the aircraft increases. This may be due the different inputs into the ACS tool in the propulsion module. While both engines are of the CFM56 engine family they of different variations of the engine. For this reason different inputs were needed. While the draw from the turbine increases the differences between these engines becomes more evident as the CFM56-7B in which the 737-M engine was modeled after burns more fuel than the CFM56-3B found on the commercial 737.

There is a difference of nearly 2000 lbs. of fuel at the high end of the engine turbine power draw. This means that for a 737-C the engine power draw is important in that it deviates further from its commercial counterpart. Since the power draw deviates the 737-M far from its original commercial variation, the subsystems that induce such a power draw are worth being examined further. The next section details a simulation tool that models the subsystems of these aircrafts to observe the flow of power from the engine itself.

4. Simulink Modeling

4.1. Introduction to the PowerFlow toolset

The PowerFlow program is a toolset developed in cooperation between Rolls-Royce and University of Illinois that models the internal performance of an aircraft. The PowerFlow toolset is modular in approach. This means that the subsystems are broken down into their own separate generic models that can be built to fit the subsystem architecture of different types of aircrafts that contain those specific subsystems [8]. This toolset breaks the subsystems downs into three main power management subsystems: electric power systems, hydraulic power systems, and thermal management systems [8].

In the PowerFlow toolset the crucial subsystems within the major subsystems are modeled such as the batteries, engine generator unit, engine generator, fans, and pumps, and many more [8]. Some subsystems such as the cabin and cockpit are modeled using dynamic behavior. Others such as hydraulics are modeled by steady-state averaged models or stochastic values [8]. These subsystems are all organized into a library of generic models so they are capable of fitting different aircraft architectures.

The library was created in order to model internal performance of the aircraft, specifically power consumption. By breaking the subsystems into pneumatic, hydraulic, and electrical components, this model is able to model the amount of power needed to operate these subsystems. It is importance to break the power consumption into subsystems in order to extract pertinent subsystem comparison within the aircraft subsystem architecture. This study will capitalize the capabilities of the PowerFlow library in order to determine the power consumption for both the commercial and military variations.

When modeling and running simulations based on systems engineering principles, the models are based off the system hierarchy, system interfaces, product structure, and functional breakdowns. It is pivotal to first outline the systems that will be included into the model and out they interact and interface with each other. The eventual model and initial figures should reflect each other exactly. Therefore, the subsystems architecture for the 737-C and 737-M are representative of the system hierarchies and system interfaces described earlier in this study. All systems and subsystems that were shown to interface with each other in the system interface diagram should be directly connected within the model.

4.2. Modeling difference for input

4.2.1. Mission GUI Input and Difference

The PowerFlow toolset is broken into two major components. The library of generic subsystem models and a mission profile graphic user interface. The mission profile GUI is where the user of the toolset build the desired mission envelope for the aircraft. The mission profile input for the PowerFlow toolset is of a higher fidelity than that of the ACS toolset so many mission legs that were approximated or combined within the ACS software are recorded with more detail of the PowerFlow mission profile GUI.

While the input for the PowerFlow mission profile input is of a higher fidelity it comes with some assumptions. Mission leg parameters such as engine thrust percentage were approximated at values slightly larger than traditional values. For the purpose of this study, the engine thrust percentage does not play a large role. The flap angles were also approximated using typical values under certain mission phases. The flap approximations for the flap angles do not make a large impact either because the current hydraulic loads are based off stochastic values of a 737 based on the type of mission phase it is in. The N/A that can be seen in the table below simply means that the PowerFlow mission profile GUI does not accept that parameter as an input and assumes trim angles. It should be noted that the mission profile inputs were kept the same for both aircrafts minus the cruise as mentioned earlier. The longer cruise time correlates to the militarized variation input.

Table 9. Mission Envelope Input for PowerFlow Simulation.

Mission Phase	Time	Engine Thrust Percentage	Altitude	Flap Angle
Startup	18 min.	0%	1500 ft.	N/A
Taxi	15 min.	10%	1500 ft.	N/A
Takeoff	50 sec.	90%	1500 ft.	12 deg.
Climb	59 min.	75%	1500 ft. – 36,700 ft.	N/A
Cruise	5.8/8.5 hr.	60%	36,700 ft.	N/A
Descent	53 min.	40%	36,700 ft. – 5000 ft.	N/A
Loiter	20 min.	60%	5000 ft.	N/A
Approach	7 min.	60%	5000 ft. – 3000 ft.	N/A
Landing	N/A	40%	3000 ft.	17 deg.
Shutdown	10 min.	0%	3000 ft.	N/A

4.2.2. Simulink Model Implementation

Now that there is a pseudo universal mission profile modeled between the ACS software and the PowerFlow GUI the subsystems now must be modeled within Simulink. Fortunately, the model for the commercial 737 has been previously built by a research team between University of Illinois and Rolls-Royce. The militarized version of the 737 was modeled by starting from the commercial model previously created and making the necessary modifications. The subsystems that will be implemented within the military variations are compiled in the table below.

Table 10. Subsystems implemented into military variations model

System	Subsystem	Actual System	Power Consumption
Radar	External Radar	MESA Radar	10.73 kW Output with 45% max eff
EWSP	SLTA	AN/AAQ-24(V)	580/1905 W (standby/maximum)
EWSP	MPTA	AN/AAQ-24(V)	300/1200 W (standby/maximum)
EWSP	SMAW	AN/AAQ-24(V)	15/55 W (standby/maximum)
EWSP	MIMS MAW	AN/AAQ-24(V)	100/140 W (standby/maximum)
EWSP	EWSP Processor	AN/AAQ-24(V)	399 W
EWSP	CIU	AN/AAQ-24(V)	25 W
Communications	Link 16 Datalink	JTIDS	1400 W
Communications	Link 11 Datalink	AN/USQ-130(V)	28 W
Communications	IDM	IDM-501	13 W
Communications	Intercommunications	ICS-150	13 W
Communications	HF	AN/ARC230/H121C	450 W
Communications	UHF	AN/ARC-187	30 W (AM) 100 W (FM)
Communications	VHF	AN/ARC-186	13 W
Communications	SATCOM	SAT-2100	13 W
Fuel System	Auxiliary Fuel Tanks	BBJ	N/A
Power Supply	MESA Power Supply	Thycon Pty. Power	Supplies MESA Radar
Mission Control	Control Consoles	C-17 Consoles	84 W

It can be seen from the table above that the only specification described for the additional subsystems is the power consumption. For this study, the power consumption is the only parameter needed to acquire the desired internal performance results. The physical size of the subsystems is not implemented into the PowerFlow Simulink model and the weights are already taken into account in the gross takeoff weight of the 737-m.

In order for these subsystems to be modeled, there must be actual systems to base the models from. While the exact subsystems implemented into the military variations of the 737 are widely available, specifications such as power consumption, were not. For this reason many of the subsystems were taken from multiple different aircraft such as the P3-C and E-3 Sentry. All of the avionics, and their specifications, were sourced from Jane's Avionics.

The EWSP subsystems were modeled after the AN/AAQ-24(V) Nemesis Directional Infra-Red Counter Measures (DIRCM) System developed by Northrop Grumman [10]. This system is planned to be implemented into the C-17 but has been assumed to be reasonable enough to model the EWSP subsystems found aboard military variations of the 737.

There was no comprehensive collection of communication subsystems for any specific earning warning military aircraft. Therefore, many of the communication systems were modeled off different communication subsystems found in various early warning aircraft or similar aircrafts. The Link 16 communication system was modeled from the Link-16 Joint Tactical Information Distribution System (JTIDS) airborne datalink terminals that can be found within the AWACS [10]. Link 11 communications were modeled by the AN/USQ-130(V) MX-512PA Link-11/TADIL-A data terminal designed by DRS technologies [10]. The improved data modem was modeled after the IDM-501 found on the Turkish AEW&C Peace Eagle [10]. Intercommunications modeled by the ICS-150 intercommunications system [10]. High frequency communication were modeled after the AN/ARC-230/HF-121C high performance radio system that can be found on the P-3 [10]. Ultra-high frequency radio communications was modeled after the AN/ARC-187 UHF Radio that can be found in the C-17 [10]. Very high frequency is modeled off the AN/ARC-186 Airborne VHF developed by Rockwell Collins and found in the C-17 [10]. Lastly, the satellite communications were modeled off the SAT-2100 communications set.

The AESA radar design is composed of “transmitter and receiver modules” (TRM) [15]. For airborne purposes a radar design typically consists of 1100-1500 TRM [15]. State of the art RF performance in “transmitter and receiver modules” is approximately 3.6 W each with a DC to RF efficiency of 16.6% [11]. Assuming an average quantity of TRM elements, it can be calculated that the power consumption for a radar similar to that of the MESA radar would be approximately 29.2 kW.

Even though the MESA radar has a large power consumption, it does not pull its power supply from the engine driven generator. An additional power supply is supplemented to deliver the main power supply for the MESA radar. Because this power supply is, in essence, a battery, this power supply was modeled by simply increasing the amount of battery modules within the PowerFlow model to supply the voltage difference needed for the MESA radar. Therefore, there was no need to create a subsystem for it that pulled such a large power load.

It can be seen that the EWSP systems have a standby and maximum power consumption. This is due to the difference in necessary functions required by the subsystems when in standby by and active mode. For simplicity, active mode was defined in the Simulink model to be when the aircraft is in cruise. When the input from the PowerFlow GUI mission profile was set to cruise the Simulink model would switch the power consumptions for the EWSP systems from standby to active.

There are 8 mission consoles found aboard the AEW&C aircraft so that is the amount of consoles that will be modeled. From the table above it can be seen that each of these consoles are 84 W each. This wattage is estimated from the specifications from the C-17 mission computer/display unit.

The PowerFlow library comes with pre-made fuel tanks and fuel pumps. In order to model the auxiliary fuel tanks additional fuel tanks and pumps were added. Two additional fuel tanks were added to represent all nine of the auxiliary fuel tanks with two fuel pumps allocated to each fuel tank. The volume of the fuel tanks were adjusted to accommodate the fuel from all nine auxiliary fuel tanks. To represent equal fuel distribution to the engines the fuel pumps distributed fuel to mixing junctions for both engines. These fuel pumps are electrically powered by the engine generators.

The PowerFlow employs a thermal management system for the passenger area of the aircraft. For the military version it is assumed that the crew would be eight personnel; one for each mission console. While for the commercial variation it is assumed to have a full aircraft. These differences are expected to have an impact on the thermal management systems. The integrated drive generators for the AEW&C is 180 kVA, which is double the commercial variation of 90 kVA.

4.3. Simulation Results

4.3.1. Simulink Mathematical Models

The mathematical models used within the PowerFlow Simulink toolset are extensive and far too in depth to cover within this study. However, this study will layout the mathematical models in which were used primarily to estimate the power consumption. These mathematical models were pulled from the PowerFlow Users Guide [14].

In order to find the power consumed by the generators seen in figures 29 and 30 we need to know the synchronous frame line voltages. The frame line voltages V_q and V_d are calculates as follows:

$$V_q = -\gamma\left(X''_d + X_{TL}\right)I_d - \left(R_s + R_{TL}\right)I_q + \gamma\left(E'_q \frac{X''_d - X_{ls}}{X'_d - X_{ls}} + \psi_{1d}\frac{X'_d - X''_d}{X'_d - X_{ls}}\right),$$

$$V_d = \gamma\left(X''_d + X_{TL}\right)I_q - \left(R_s + R_{TL}\right)I_d + \gamma\left(E'_d \frac{X''_q - X_{ls}}{X'_q - X_{ls}} - \psi_{2q}\frac{X'_q - X''_q}{X'_q - X_{ls}}\right),$$

Where X_d and X_q are the direct and quadrature axis per-unit reactances, respectively; X'_d and X'_q are the direct and quadrature axis per-unit transient reactances, respectively; X''_d and X''_q are the direct and quadrature axis per-unit subtransient reactances [14]. R_{TL} and X_{TL} are the transmission line per-unit resistance and reactance, respectively; γ is the per-unit electrical frequency (typical base value is 377 rad/s), R_s is the synchronous machine per-unit stator resistance [14].

These two synchronous frame line voltages are then muxed to create the synchronous machine line voltage V_{dq0}. This voltage is calculated for the synchronous generator and the exciter which are both subsystems within the generator unit. This voltage is then multiplied by the generator input current to calculate the power of the generator unit. It should be noted that these method of power consumption is used in both generator units and the APU. The top level equation is simply:

$$P_{generator} = V_{dq0} * I_{in}$$

Power loss within the generators and APU are also accounted for as power consumption. The equation to calculate the power loss within the generators is as follows:

$$P_{loss} = S_B \left(\gamma T_g - E_d I_d - E_q I_q \right)$$

Where $S_b$is the base power, T_g is the per unit time input of torque to generator from gearbox, where the various I and E are frame line current and voltages respectively [14]. This is the only power consumption totaled for the electrical system. This is because all other electrical components such as the various avionics detailed above are funneled into a load which is then fed into the generators. To give an idea about how the power was calculated at the lower level of the generator a snapshot was is shown in figure 29 from [14] to visualize how power was calculated in this study.

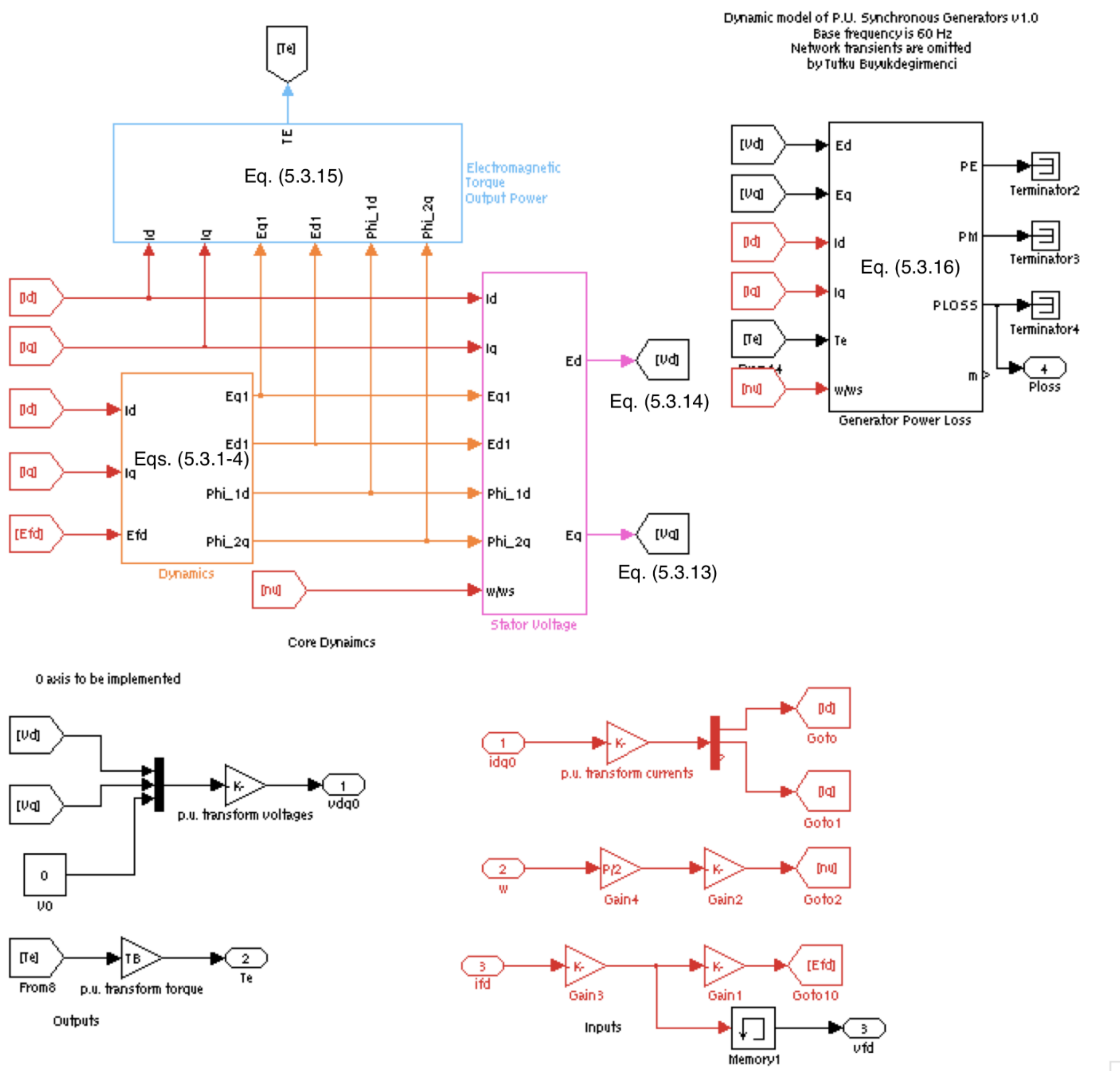

Figure 29. Generator system power calculation

The power consumed by the hydraulic systems is given by the following equation:

$$P = \dot{m}\tilde{P}/\rho$$

Where $\dot{m}$ is the mass flow rate of the hydraulic fluid, $\tilde{P}$ is the fluid pressure, and ρ is the fluid density. This power is calculated for every control surface in which hydraulics are used.

The power consumed for the pneumatics are drawn from the PACK systems. This power comes from the compressor power consumption. In this calculation it is assumed the process is adiabatic and 100% efficient. Given those assumptions the power in each pack is calculated using the following equation:

$$P = \dot{m} C_p (T_{bleed} - T_{cabin})$$

Where $\dot{m}$ is the mass flow rate of bleed air through the pack. $C_p$is the specific heat constant at constant pressure for air. The temperature differential is the difference in temperature between the bleed air and the desired cabin temperature.The power is directly related to the difference of temperature that the pack is required to satisfy.

4.3.2. PowerFlow Simulation Results

Now that the models and mission profiles for both of the aircraft variations have been developed the models must be simulated. Each model was simulated separately. The models are complex and thus take a large amount of computing power. Each simulation took approximately 8 hours on an Intel Core i3-3220 CPU @ 3.30 GHz with 8 GB RAM. Power consumption among the subsystems were broken down between pneumatic, hydraulic, and electrical components. Pneumatics encompasses the left and right packs. Hydraulics encompasses the actuation loads and the electrical components encompass avionics, generator losses, and other electrical component losses. The results are as follows.

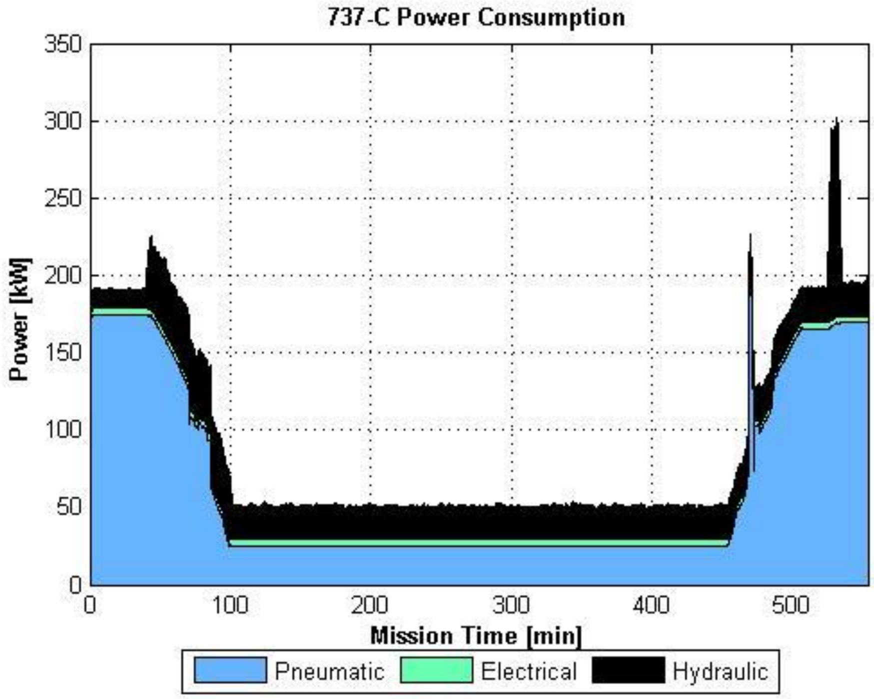

Figure 30. 737-C Power Consumption Results

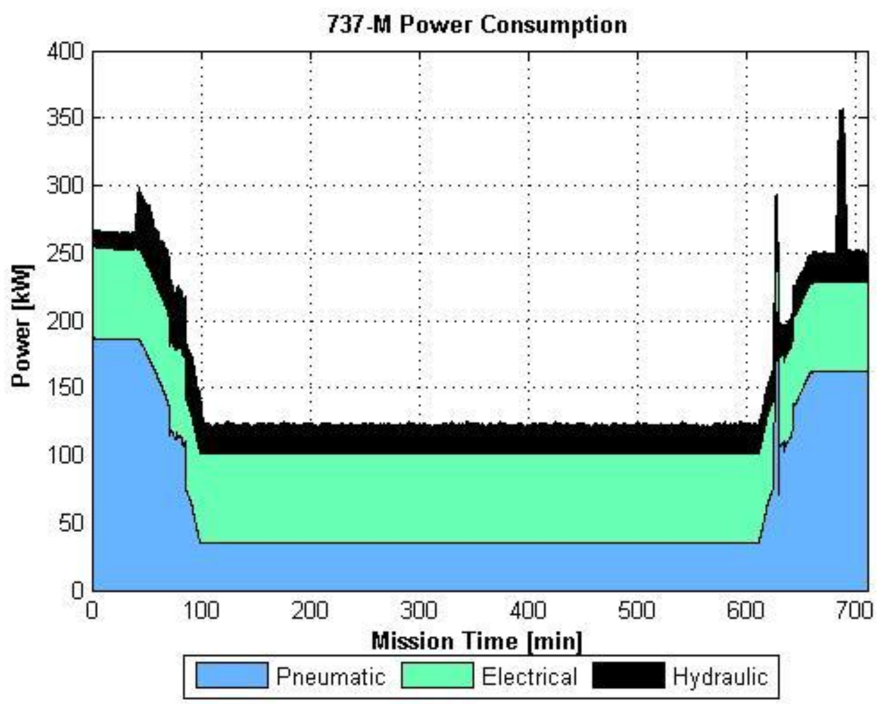

Figure 31. 737-M Power Consumption Results

At first glance, it is palpably evident that the military variations draws more power for electrical purposes than that of its commercial counterpart. This is due to the additional avionics subsystems and additional fuel pumps. As expected, the cruise portion of the mission yields a higher percentage of electrical subsystems power consumption due to EWSP systems being switched to active mode.

It can also be seen that the pneumatic systems for the military variation is also slightly higher. The PowerFlow toolset does employ a heat model for heat given off by humans but also by the subsystems themselves. This pneumatic system could be consuming more power due to additional heat added to the closed system from the additional subsystems.

This power consumption output is outputted as an array based on real time. That is an array that's length is equal to the number of seconds of the mission profile. To get a better idea of the power consumption a time weighted average power consumption is taken from the array. The array was broken up into segments for each mission leg and averaged. The results are compiled below. Since these results will later be used within the ACS software, which has the aforementioned limitations to the mission profile build, these results are broken into start to takeoff, climb, cruise, descent, and loiter.

Table 11. Total Power Consumption Weighted Averages

Mission Leg	Commercial Variation Average	Military Variation Average
Start to Takeoff	191.6 kW	265.5 kW
Climb	160.7 kW	234.3 kW
Cruise	50.89 kW	122.6 kW
Descent	88.39 kW	203.6 kW
Loiter	193.93 kW	250.9 kW
Total Average	94.68 kW	155.7 kW

Table 12. Military Variation Power Consumption Breakdown

Mission Leg	Pneumatic Power Average	Hydraulic Power Average	Electrical Power Average
Start To Takeoff	186.7 kW	12.69 kW	66.28 kW
Climb	125.7 kW	42.06 kW	66.52 kW
Cruise	34.89 kW	21.45 kW	66.27 kW
Descent	115.87 kW	21.55 kW	66.15 kW
Loiter	162.16 kW	22.67 kW	66.14 kW
Total	65.77 kW	23.69 kW	66.27 kW

Table 13. Commercial Variation Power Consumption Breakdown

Mission Leg	Pneumatic Power Average	Hydraulic Power Average	Electrical Power Average
Start To Takeoff	174.7 kW	12.69 kW	4.23 kW
Climb	114.1 kW	42.06 kW	4.53 kW
Cruise	25.07 kW	21.47 kW	4.35 kW
Descent	62.54 kW	21.59 kW	4.26 kW
Loiter	165.5 kW	23.51 kW	4.26 kW
Total	65.97 kW	24.38 kW	4.33 kW

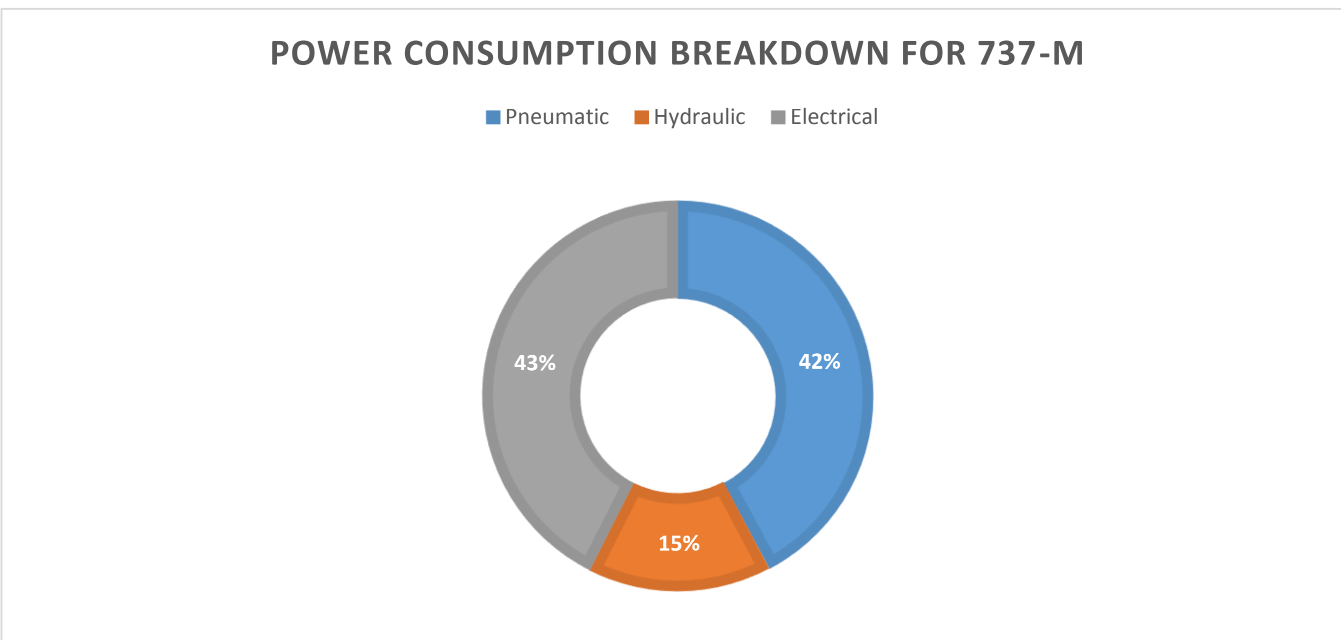

Figure 32. Power Consumption Breakdown For 737-M

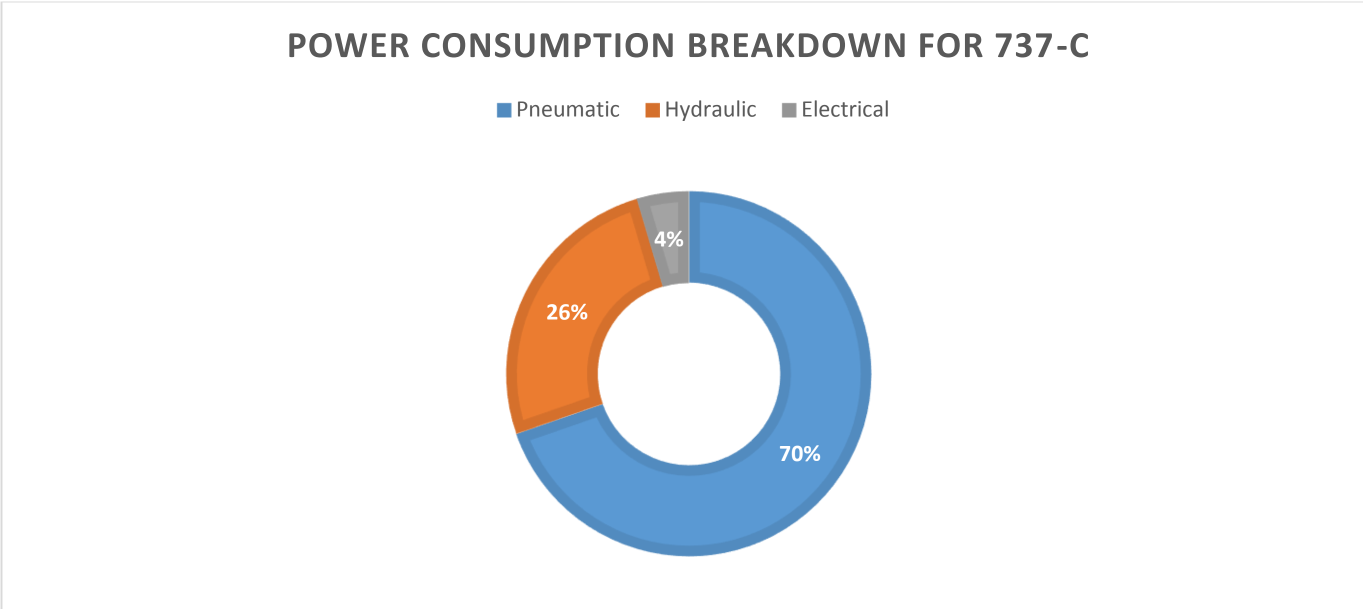

Figure 33. Power Consumption Breakdown For 737-C

These results both confirm initial thoughts and offer surprising insight. As mentioned earlier the power generated by the commercial and military variations generators are 90 kVA and 180 kVA respectively. As can be seen by the simulation results the commercial variation resulted in a power consumption average of 94.68 kW which is just higher than its design point. This means that, on average, the generators are running slightly over 50% of their maximum output. This is to be expected since each engine has a 90 kVA generator. In the case of an engine or generator fault the aircraft would be able to maintain all operating functions.

The military variation resulted in an average power consumption of 155.7 kW which is considerably less than its 180 kW design point. This could be for many reasons. There could be subsystems that were not implemented at all into the model or modeled differently than their actual subsystems. As was mentioned earlier in the study, many of the subsystems for military variations of the 737 such as the AWACS, AEW&C, and the P-8 Poseidon all have roughly the same subsystems but with different specifications. Information regarding these subsystems was scarce thus specifications from various subsystems among these aircraft were used within the model. This estimation could have led to error within the model.

It can be seen from comparing the hydraulic power average between table 9 and table 10 the hydraulic loads are almost the same between aircraft variations. This is what to be expected from the model since the hydraulic loads in the PowerFlow simulation are based off stochastic hydraulic loads from a 737. This is intuitive also because the aircrafts actuation systems are inherently identical and will, in turn, lead to similar results. While this is what is to be expected there are limitations within the PowerFlow toolset that could have led to more consistent results. The fidelity of the PowerFlow hydraulic load modeling could be higher since it is currently just based of stochastic loads rather than a full hydraulic model.

It can be seen that the pneumatic loads are roughly the same for both aircraft variations. But at a closer look, it can also be seen that every leg besides loiter yields a higher pneumatic power consumption. Yet, there is observed an almost equal pneumatic loading. This is because of the longer length of the cruise leg for the military variation. Since this cruise leg power consumption is significantly less than the average for the entire mission it’s longer time length gives a greater impact on overall average. If the cruise length was to be shorter than the defined mission profile, then the overall average power consumption for the military variation would have resulted to be larger than the commercial version; and vice-versa.

It is surprising to see such a sharp increase in pneumatic power consumption during the loiter phase of the mission. This could be due to an error in the simulation or it could be due to cruising as atmospheric conditions of low altitude for an extended period of time.

The electrical power consumption is where the two simulations showed the most difference and it is intuitive since most of the additional subsystems added to the PowerFlow model for the military variation are electronic avionics. Military aircrafts require EWSP systems and more powerful communications systems so this large jump in electrical power consumption is reasonable.

When comparing the two pie charts in figure 32 and 33, it can be seen that electrical power consumption for the 737-C is only 4% of the total consumption where, in contrast, is 43% for the 737-M. This 39% increase in total power consumption for the electrical components alone is a rather large increase but also indicates that the internal performance of the 737-M has room for improvement in regards to its electrical consumption.

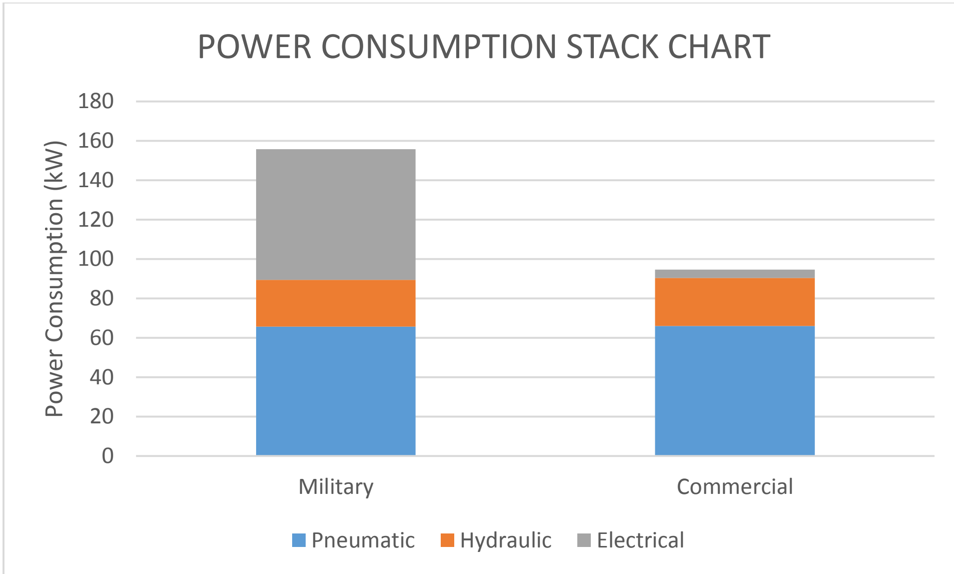

Figure 34. Power Consumption Stack Chart

The pie charts in figure 32 and 33 give a visual for the percentages of power consumption for each subsystem but it does not highlight the amount of power used for each subsystem that can be seen in the stack chart above. Figure 34 gives a visual representation of the system breakdown magnitudes. The pneumatic and hydraulic systems are not exactly the same, however, the greatest magnitude is clearly the electrical systems. This is due to the large amount of avionics that are included on the 737-M.

4.3.3. PowerFlow/ACS Hybrid Results

Now that the internal performance have been indicated, it is important to see how this can affect the overall performance of the aircrafts. This study extracts the internal performance results from the PowerFlow simulation and uses the results within ACS to determine the impact internal performance has on the entire system.

An algorithm was designed to run the PowerFlow simulation and extract the averaged power consumption for each mission leg that can be found in table 8. As mentioned earlier in the study, the ACS input comes with an input for the power draw from the inner turbine. This power draw from the inner turbine was then replaced by the average for the first mission leg and the ACS software was ran. The amount of fuel used for that specific mission leg was then recorded. This was iterated for each mission phase. This bypasses the limitations of the ACS software and allows for a variable power draw from the engine. While this may not accommodate changes in the power draw within the mission phases themselves it does create a higher fidelity result than if the power draw was held constant. Figure 35 below gives a visual representation of the algorithm used.

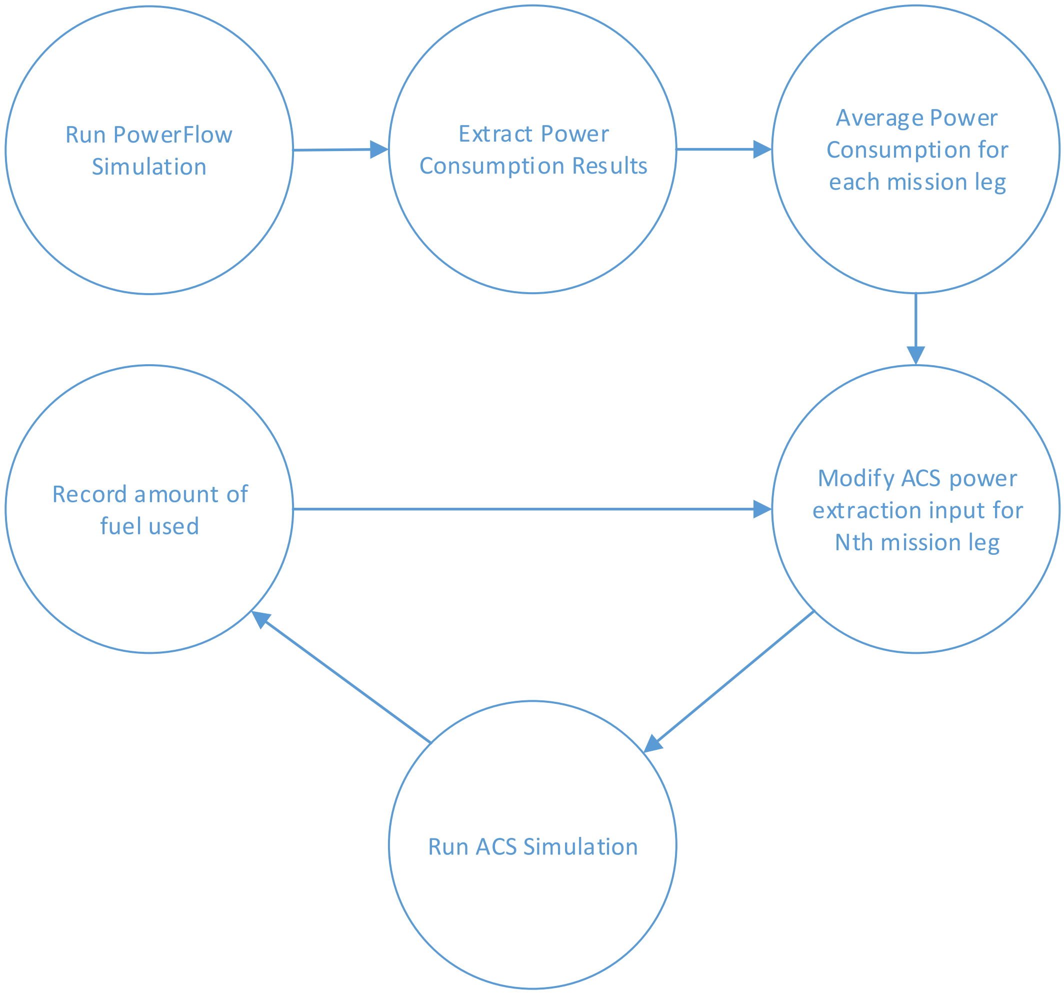

Figure 35. Visual representation of PowerFlow to ACS algorithm

It is important to note that the mission envelopes for both aircraft variations were designed to be the same within the ACS software. This was done in order to obtain results that was able to compare the fuel differential between the aircraft variations under the same conditions. The average power consumptions were used from the previous study. It should also be noted that the ACS software assumes an immediate descent and thus no distance is covered in the descent. This is a large limitation in the results but is something limited by the software.

It should be noted that this limitation is the same for both aircraft variations so as to receive consistent results. Running the algorithm produces the following results:

Table 14. Fuel use breakdown

Mission Leg	737-M	737-C
Start To Takeoff	2531 lbs.	1102 lbs.
Climb	7859 lbs.	7361 lbs.
Cruise	47951 lbs.	48514 lbs.
Loiter	2814 lbs.	1983 lbs.
Total	61155 lbs.	58960 lbs.
Differential	2195 lbs.	

These results are very important because it encompasses a crucial aircraft performance parameter: fuel economy. Both mission profiles has the exact same characteristics as the mission profiles used in previous segments of this study but the cruise leg of this mission profile was set to 3000 nmi ferry for each aircraft. That is for a 3000 nmi ferry mission the military variation burns approximately 2,195 lbs of fuel more than its commercial counterpart.

With such complex systems there may be many contributing factors that could lead to this fuel disparity. One possible cause could be the addition of the avionics subsystems that pull a lot of power from the engine. As can be seen from figures 32 and 33, the 737-M has a large amount of electrical power consumption in comparison to the 737-C and, in turn, could have been a leading contributor to this fuel differential.

Another possible cause could be that the 737-M simply has a higher drag than the 737-C. This was proven using the ACS software tool earlier within this study. Fundamental force diagrams show that a higher drag would, of course, need a higher thrust to counter the opposite force and maintain the desired velocity. The velocity for all mission legs were kept the same between ACS inputs. Thus to maintain this equal velocity at a higher drag the 737-M must burn fuel at a higher rate.

That being stated, the results of this experiment are mainly intuitive besides the results of the cruise mission leg. It can be seen that the amount of fuel burned for the 737-C is greater than the 737-M. This is counter-intuitive of what would be expected given the narrative of the previous points that the 737-M has a higher drag and draws more power from the engine. These results are no doubt surprising but can be speculated further in figure 36 below.

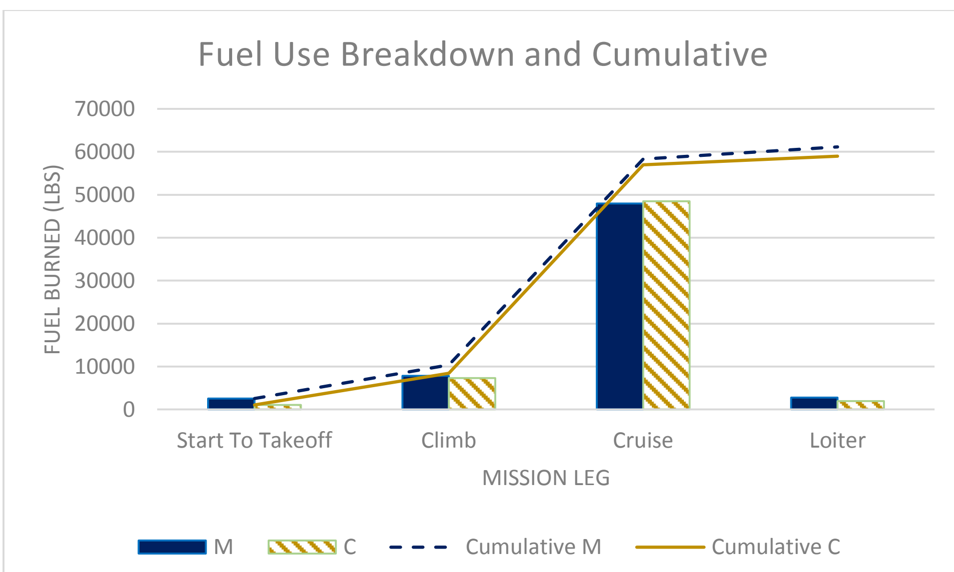

Figure 36. Mission leg fuel breakdown with cumulative values

The cruise mission leg is, by far, the longest mission leg conducted within this experiment and it also the only mission leg in which the 737-M results in a lower consumption of fuel. This may indicate a few factors. This can indicate that at a steady state altitude the 737-C burns a very low amount of fuel more than the 737-M. This can be due to the internal environmental control systems. The PowerFlow model is built as closed loop system. This means that no heat enters or leaves the boundary of the simulation. The amount of passengers for the 737-C is set to full capacity while the 737-M is set to a full mission crew of 8. The PowerFlow and ACS software models do take into account the amount of humans aboard the aircrafts as they do emit heat themselves.

At an altitude of 36,000 feet the environmental control systems is working at full duty to maintain the temperature and pressure within the fuselage. This disparity in humans within the aircrafts could lead to higher fuel burns for the 737-C over a long mission leg at such a high altitude. This is only a possible explanation and does not fully describe the root of the results. The military variation does, as shown earlier in the study, have a higher drag than the commercial variation, thus it should be expected that in any mission leg the fuel consumption should be higher for the military variation. However, these results show that they are very compatible. While the increased drag of the military variation does make a fuel consumption difference, it can be seen from figure 36 that, in a cumulative sense, the drag makes a small difference in the bigger picture.

Now that we know exactly how much fuel it costs for each mission leg, it would be beneficial to visualize the percentage of total fuel used for each mission leg. Below are figures that represent the fuel breakdown percentages.

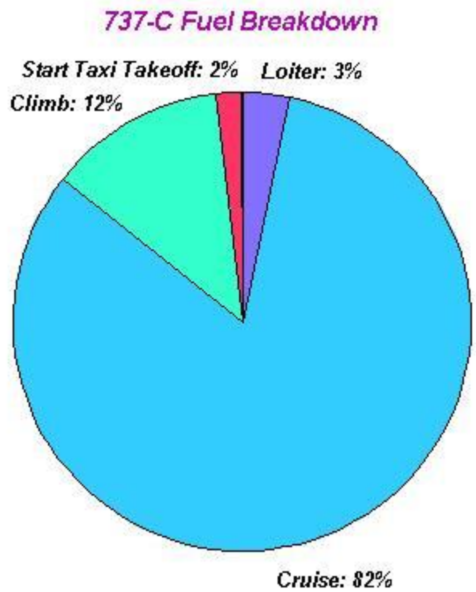

Figure 37. 737-C Fuel Pie Chart Breakdown

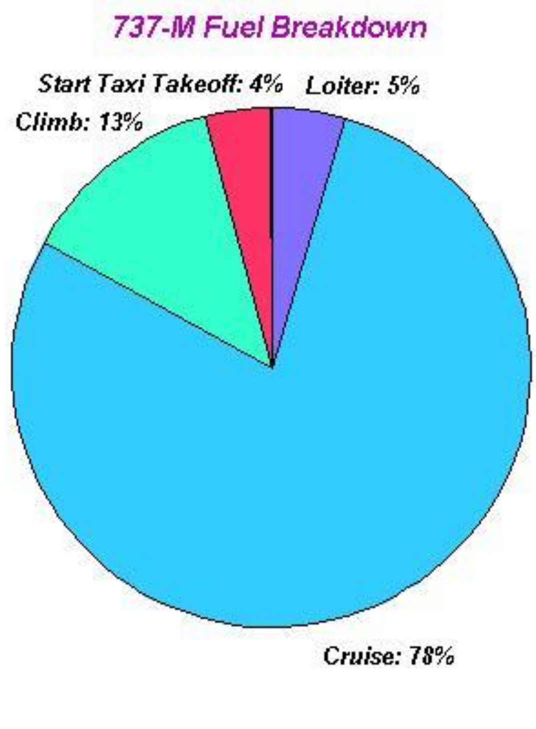

Figure 38. 737-M Fuel Pie Chart Breakdown

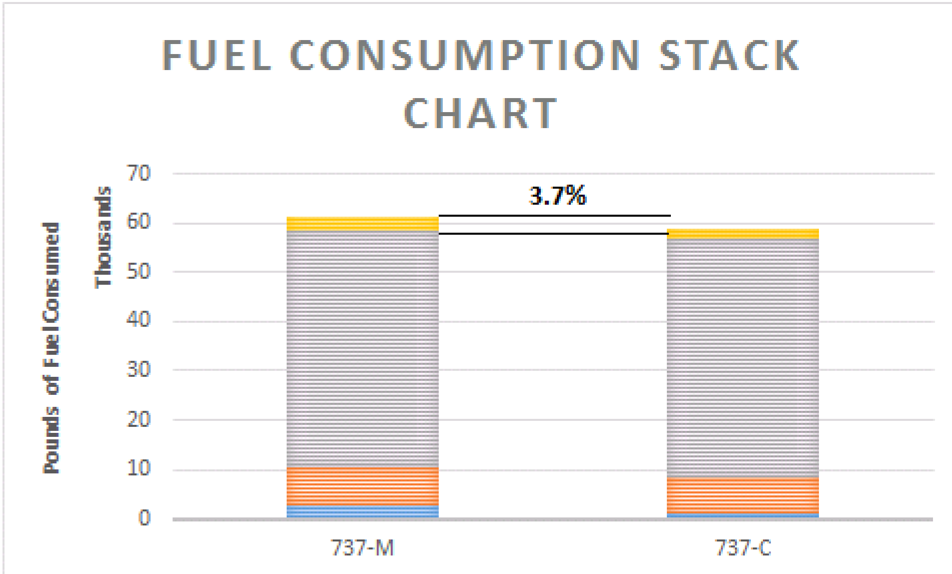

Figure 39. Fuel Consumption Stack Chart

There may be several reasons why one would want to break the mission profile up by fuel used for each mission leg. One highly researched area almost all vehicles is fuel optimization. When the 737 was designed it was designed around requirements; many of which were operation requirements that encompasses fuel consumption. When many of the early warning aircrafts were developed from the 737 aircraft, it was meant to accomplish an entirely new set of operational requirements. These operational requirements stray from that of the commercial 737 and thus the early warning aircraft was never designed, at an early stage, around its operational requirements. It is important to break the fuel consumption into the mission legs because it shows what type of operations would best fit such as a military variation of the 737. It could also help design engineers and system architects conduct trade studies and experiments on how to improve the performance of fuel based of this breakdown.

It can be seen from the pie charts in figure 37 and 38 that the start to takeoff fuel consumption for the 737-M is 4% while the 737-C is 2%. This could be due to the higher gross weights of the 737-M. Again what has been surprising throughout these results is the cruise fuel consumption. Looking at the charts above 78% of fuel used is used within cruise for the 737-M and 82% for the 737-C. Reiterating what was stated earlier, this shows a higher fuel efficiency for the 737-M. Since the pie charts do not give an accurate visual representation on how much fuel was spent for the mission a stack chart was shown. This stack chart does put the amount of extra fuel used in perspective. While 2,195 lbs. of fuel would seem to be a lot of fuel, it can be seen by the stack chart above that that is only a small percentage of the total fuel burned.

Start to takeoff, climb, and loiter are segments for the 737-M that have a larger fuel consumption percentage than the 737-C and may have room for improvement. Since the cruise mission leg has a smaller fuel percentage, it is indicative that the concept of operations for a military variation should be based around missions with very long cruise lengths and minimal altitude changes. This type of concept of operations would lead to a higher fuel performance of the aircraft. It is pivotal to note that this a conclusion based solely off the results recorded from this experiment. These results incorporate many assumptions and simplifications that may have screwed results.

5. Conclusions and Future Work

5.1. Conclusions

By using a systems engineering modeling and simulation approach it was possible to determine the differences between two similar complex systems and conduct an analysis as to why these difference were observed based on the differences within the subsystem architecture.

Many different studies were conducted after the two systems were broken down into their. The intention was to compare and contact the results given the difference in subsystem architecture. Some important arguments from this study include:

- ACS aerodynamics show a clear increase in overall drag for the military variation.
- Increasing power draw from the engine shows a large increase in fuel consumption; a maximum of 3.48% for the military variation and 2.55% for the commercial variation.
- Additional power draw from the engine in the military variation is mostly from electrical power consumption; increasing electrical power consumption from 4% in the commercial variation to 43% in the military variation.
- The military variation consumes roughly 3.7% more fuel for the same mission profile
- Military variation concept of operations operates efficiently with mission profiles that consist of long cruise lengths.

This study was conducted to demonstrate that using a complete system breakdown, one can fragment a complex system and conduct in depth analysis. Using the results of these experiments many of the results could be attributed to certain systems since the higher level (level 0 aircraft systems) was broken down into something more manageable. This allows the ability to simplify results and attribute outcomes to certain subsystems.

5.2. Future Work

As previously stated, there are many assumptions and simplifications taken into account throughout this study. As mentioned, the hydraulic loads within the PowerFlow model are stochastic hydraulic loads pulled from recorded data from 737 actuation systems. The PowerFlow library and model is currently a work in progress and its fidelity continues to improve with time. This study could possibly be conducted again once the fidelity of the hydraulics improves to give a more accurate presentation for of power consumption.

The scope of the mission profile for the ACS input has been said to be very limited. There were many mission legs that were could not be modeled due to the limitations of the software. One large hindrance to the study was that the ACS software assumed an immediate drop in altitude during the descent phase. Because of this assumption, the study could not include the descent phase in any of the fuel consumption analysis. However, in future studies, this could be navigated. This could be circumnavigated by modeling the descent phase using multiple segmented cruises at decreasing altitudes. Just as in calculus, the more steps there are the more accurate the model. But due to the uncertainty of multiple cruises being able to model a descent phase, this was omitted from the study but could be implemented in future work.

Lastly and very importantly is verification and validation of the PowerFlow model. Verification makes sure the model has been implemented correctly while validation shows that the models accurately represents the real life system. It is assumed that the ACS software model has already been verified and validated since it is a quasi-commercial product. However, the PowerFlow model is currently a research work in progress. It can be argued that the model has been verified as many components of the models have been examined by experts in the area. However, the model has not been validated and is currently a task being currently investigated by the research team behind the PowerFlow project. It is important to validate the PowerFlow model as it will give the results of this study more certainty.

References

[1] ACS/ACSYNT User's Guide, Avid LLC, 2013.

[2] Brady, Chris. The Boeing 737 Technical Guide. U.K.: Tech Pilot Services, 2007. Print.

[3] Shevell, Richard Shepherd. *Fundamentals of Flight*. Englewood Cliffs, NJ: Prentice-Hall, 1983. Print.

[4] Carlson, John. 'Prediction of Very High Reynolds Number Compressible Skin Friction'. NASA Langley Research Center, Hampton, Virginia.

[5] Boeing, 'Boeing Business Jets Auxiliary Fuel Tank System Certified'. 1999. [Online]. Available: http://boeing.mediaroom.com/1999-05-21-Boeing-Business-Jets-Auxiliary-Fuel-Tank-System-Certified.

[6] Aviation Weekly Intelligence Network. 'Gas Turbine Engine Specification Details'. [N/A].

[7] CFM56-7B Turbofan Engine. [Online]. Available: http://www.cfmaeroengines.com/engines/cfm56-7b

[8] Williams, M., et al. 'PowerFlow: a Toolbox for Modeling and Simulation of Aircraft Systems'. University of Illinois. 2015.

[9] 737 Flight Crew Operations Manual, The Boeing Company. 2000.

[10] E. Downs, *Jane's Avionics*, Janes Information Group, 2007.

[11] Clark, William. 'Millimeter Wave Seeker Technology'. AIAA. 1992. Print.

[12] Haupt, Randy. Antenna Arrays a Computational Approach. IEEE Press. 2010.

[13] Anand, G, et al. 'Analysis, and Evaluation of a Propulsive Wing Concept'. [AIAA]. 2016.

[14] PowerFlow Users Guide, Rolls-Royce, 2013.

[15] Kopp, Carlo. 'Evolution of AESA Radar Technology'. Microwave Journal. 2012. [Online]. Available: http://www.microwavejournal.com/articles/17992-evolution-of-aesa-radar-technology?v=preview.

Appendix A – ACS Input Files

```
$DATA BLOCK A
BOEING 737-700 TRANSPORT
$DATA BLOCK B
1

$DATA BLOCK V
END
TRANSPORT
    6    4    6 1830 1845    0    0    0    2    1    7    0
    .0002       .6    10e8
    1    2    3    4    6    9
    1    2    6   -9
    1    3    4    2    6    9
 ***** GEOMETRY FOR BOEING 737-700 *****
 $WING
 SWEEP=25.02,
 KSWEEP=1,
 TAPER=0.159,
 TCROOT=0.12,
 TCTIP=0.12,
 ZROOT=-01.0,
 DIHED=6.0,
 AREA = 1340.97,
 AR = 9.45,
 WFFRAC=0.39,
 XWING=0.45,
 $END

 $HTAIL
 SWEEP=30.0,
 KSWEEP=1,
 AR=6.16,
 TAPER=0.203,
 TCROOT=0.1,
```

```
TCTIP=0.1,
ZROOT=0.3,
AREA=352.84098,
SIZIT=F,
XHTAIL=0.98,
$END

$VTAIL
SWEEP=30,
KSWEEP=1,
AR=1.91,
TAPER=0.271,
TCROOT=0.1,
TCTIP=0.1,
ZROOT=.6,
XVTAIL=.97,
AREA=284.59779,
SIZIT=F,
$END

$FUS
FRN=2.18,
FRAB=3.5,
WALL=0.4,
DRADAR=0.0,
LRADAR=0.0,
BODL=105.57743,
BDMAX=12.33596,
WFUEL = 45856.151,
$END

$FUEL
DEN = 49.5,
FRAC = 1,
WFUEL = 45856.151,
```

```
 $END

 $ENGINE  N=2,   $END
 $CREW NCREW=2, $END
 $ELEC LENGTH=3.0, $END
 $ELEC LENGTH=2.0, $END
 $PASS NFIRST=0, NCOACH=115, SEATWC=20.0, $END

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
 $TRDATA
  TIMTO1=33,
  TIMTO2=0.0,
  FRFURE=0.14,
  RANGE=3261.339,
  WFEXT=0.0,
  WFTRAP=0.0,
  XDESC=100.0,
  CRMACH=.75,
  IPSTO1=5,
  IPSTO2=2,
  MMPROP=1,
  WFUEL = 45856.151,
  IPSIZE=0,
  NLEGCL=20,
  !LEGRES=3,
  IBREG = 1,
  NMISS = 1,
  NCRUSE=1,
 $END
       5
        MACH NO.   ALTITUDE     HORIZONTAL    NO.  VIND
PHASE   START END  START  END    DIST  TIME   TURN  "G"S  WKFUEL M IP IX W B A P
------- ---- ----  ----- -----  ------ -----  ---- -----  ------ - -- -- - - - -
CLIMB   0.20   -1      0  1500    -1.0   0.0   0.0   0.0  3.0000 1  2 -1 0 0 0 0
CLIMB     -1 0.77   1500 36700    -1.0   0.0   0.0   0.0  3.0000 1  2 -1 0 0 0 1
```

```
CRUISE  0.77 0.77     -1 36700  2600.0   0.0   0.0   0.0  1.0000 1  4  0 0 0 0 0
DESCENT 0.77 0.40     -1  1500     0.0   0.0   0.0   0.0  1.0000 1  4  0 0 0 0 0
LOITER  0.40 0.40     -1  1500     0.0  20.0   0.0   0.0  1.0000 1  4  0 0 0 0 0

***** AERODYNAMICS OF THE BOEING 737-700 *****
 $ACHAR ABOSB=0.18,  ALMAX=14.0, AMC=40.0, BDNOSE= 0.0, BTEF=0.00,
        CLO=0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
        CLOC=0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
        CLOW=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
        CMO=0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
        MACHN=0.75, RALOIT=  0.00, RCLMAX=1.000,
        ROC=.05, ROCAN=.020, SFWF=1.00, SMNDR=0.80,
        SMNSWP=0.0,0.76,0.80,0.82,0.84,0.85,0.86,0.87,0.89,0.91,
        SPANAC= 0.0, SWPMAX=60.0, SWPMIN= 0.0, XCDC=0.60, XCDW=0.60,
        YSWP= 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
        ALELJ=5, INORM=1, ISMNDR=0,
        ISUPCR=0, ITRAP=0, IXCD=1,
        ELLIPC=F, ELLIPH=F, ELLIPW=F, $END
 $AMULT CSF=0.000, ESSF=0.000, FCD=1.00, FCDF=0.975, FCDL=1.0,
        FCDRA=10*1.0,
        FCDO=1.0, FCDW=.500, FCDWB=1.000, FENG=1.000, FINTF=1.000,
        FLBCOR=1.000, FLECOR=1.000, FMDR=0.95, $END
 $ATRIM CAND= 0.0, CFLAP=0.20, CGM=0.25,
        FLDM= 10*1.0,
        FVCAM=10*.75,
        SPANF= 0.75, ZCG= 0.0,
        ITRIM=10*1, IVCAM=1, $END
 $ADET  ALIN= -1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
        ALTV= 8*35000.0,
        CLINPT=0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
        SMN=0.4,0.45,0.5,0.55,0.6,0.65,0.70,0.75,
        ICOD=1, IPLOT=1,
        ISTRS=0,0,0,0,0,0,0,0,0,0,
        ITB=0,0,0,0,0,0,0,0,0,0,
        ITS=0,0,0,0,0,0,0,0,0,0,
```

```
      NALF=10, NMDTL=10, $END
 $ADRAG CDBMB= 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
      CDONPT= 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
      CDSTR= 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
      CDTNK= 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
      CDEXTR= 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      SMNCDO=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
      SMNBMB=0.00,0.20,0.40,0.60,0.80,1.00,1.20,1.40,1.60,1.80,
      SMSTRS=0.00,0.20,0.40,0.60,0.80,1.00,1.20,1.40,1.60,1.80,
      SMTANK=0.00,0.20,0.40,0.60,0.80,1.00,1.20,1.40,1.60,1.80,
      SMEXTR=0.00,0.20,0.40,0.60,0.80,1.00,1.20,1.40,1.60,1.80,
      ICDO=0, $END
 $ATAKE CLLAND=-1.00, CLTO=-1.00, DELFLD=45.0, DELFTO=10.0,
      DELLED=30.0, DELLTO=15.0, LDLAND=-1.0, LDTO=-1.0, ALFROT=6.,$END
 $APRINT ECHOIN=1, ECHOUT=0, INTM=0, IPBLNT=0, IPCAN=0, IPENG=0, IPEXT=0,
      IPFLAP=0, IPFRIC=0, IPINTF=0, IPLIFT=0, IPMIN=0, IPWAVE=0, KERROR=0,
      $END
***** GENERAL ELECTRIC CFM56-7B20 TURBOFAN *****
6
 $LEWIS TWOAB=20600.,
          AENDIA=5.08,
          AENLE=8.225,  !These values come from reference 15
          AENWT=5234.,
          BA=5.5,
          DIA1=3,
          P2P1=22.7,
          ETAC1=.92,    !Typical efficiencies for aircraft engines
          ETAF1=.95,
          ETAT1=.95,
          T3=2800.,
   XMACH=0.0,0.6,0.65,0.70,0.75,0.85,
          ALTD=0.,5*25000.0,
          MACH1=0.85,
          SFSFC1=0.9,
          AuxPowerInner=193.3,
```

```
 $END

 $INLET
  LM=10.,
  SFPRFP = 1.0,
  NINL  =2,
 $END
 $AFTBD    $END
TRANSPORT
***** 737-700 WEIGHTS *****
 $OPTS   ITAIL=1, WGTO=154500.0, AFMACH=.80, ITHRV = 4, $END
 $FIXW   $END
 $CMAN
  FEE=0.190,            NNG=1,        RTRTM = 12.,    CTJI = 0.,
  NVEH= 1., 200., 400.,   NDATA=3,       NV=504.,        NFV=1.0,
  RE=65.0,              RT=55.0,       YEAR = 1992.,   API = 0.08,
  LEARNA1=81.5,          LEARNP1=100.,  LEARN1=81.5,    FENGQ =2500.,
  RATE = .667,1.,3.,3.,3.,3.,3.,0.,0.,0., RTRTA = 12.,
  PUNITS = 100.,
  ICONFG=6,     IPROD=1,        IENGS = 1,     IOPS=1,   XFASSY =.05,
  ICSHFLW=1,    IAIRROI=1,      CFENG=1.00,    ENSPAO=.23,  $END
 $COPER
  AMT=0.,       COFL=.10742,    COIL=2.63,     NSL=2,       RCH=500.,
  RESDVL=10.,   RINRST=8.,      DWNPYM=0.,     RL=25.0,     RCSL=3000.,
  DESS=300.,    FINSUR=.35,     U=5000.,       H=32000.,    BDMAIN=200.,
  SL=4600.,4600.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,            ECLIFE=20.,
  FLF=.65,.65,.65,.65,0.,0.,0.,0.,0.,0.,0.,0.,             TXRATE=34.,
  CLF=.65,.65,.65,.65,0.,0.,0.,0.,0.,0.,0.,0.,             GRNDTM =1.,
  DPT=1.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.,0.,
  AI1=.5, AI2=5.2, AI3=215., AI4= 63.55, AI5=2.0, AI6=11., AI7=80.,
  AI8=.0214, AI9=.012, AI10=.0703, AI11=11., AI12=38.,
  $END
 $CMAINT
 $END
```

```
$DATA BLOCK A
AEW&C TRANSPORT
$DATA BLOCK B
1

$DATA BLOCK V
END
TRANSPORT
    6    4    6 1830 1845    0    0    0    2    1    7    0
    .0002       .6    10e8
    1    2    3    4    6    9
    1    2    6   -9
    1    3    4    2    6    9
 ***** GEOMETRY FOR AEW&C *****
 $WING
 SWEEP=25.02,
 KSWEEP=1,
 TAPER=0.159,
 TCROOT=0.12,
 TCTIP=0.12,
 ZROOT=-01.0,
 DIHED=6.0,
 AREA = 1340.97,
 AR = 9.45,
 WFFRAC=0.39,
 XWING=0.45,
 $END

 $HTAIL
 SWEEP=30.0,
 KSWEEP=1,
 AR=6.16,
 TAPER=0.203,
 TCROOT=0.1,
 TCTIP=0.1,
```

```
ZROOT=0.3,
AREA=352.84098,
SIZIT=F,
XHTAIL=0.98,
$END

$VTAIL
SWEEP=30,
KSWEEP=1,
AR=1.91,
TAPER=0.271,
TCROOT=0.1,
TCTIP=0.1,
ZROOT=.6,
XVTAIL=.97,
AREA=284.59779,
SIZIT=F,
$END

$FUS
FRN=2.18,
FRAB=3.5,
WALL=0.4,
DRADAR=0.0,
LRADAR=0.0,
BODL=105.57743,
BDMAX=12.33596,
WFUEL = 71316.151,
$END

$FUEL
DEN = 49.5,
FRAC = .5,
WFUEL = 71316.151,
$END
```

```
$FUEL
DEN = 49.5,
FRAC = .5,
WFUEL = 71316.151,
$END

$ENGINE  N=2,   $END
$CREW NCREW=8, $END
$ELEC LENGTH=3.0, $END
$ELEC LENGTH=2.0, $END
$PASS NFIRST=0, NCOACH=0, SEATWC=20.0,
$END



$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$TRDATA
 TIMTO1=33,
 TIMTO2=0.0,
 FRFURE=0.14,
 RANGE= 3801.296,
 WFEXT=0.0,
 WFTRAP=0.0,
 XDESC=100.0,
 CRMACH=.40,
 IPSTO1=5,
 IPSTO2=2,
 MMPROP=1,
 WFUEL = 71316.151,
 IPSIZE=0,
 NLEGCL=20,
 !LEGRES=3,
 IBREG = 1,
 NMISS = 1,
 NCRUSE=1,
```

```
 $END
      5
       MACH NO.   ALTITUDE     HORIZONTAL    NO.  VIND
PHASE   START END  START  END    DIST  TIME   TURN  "G"S  WKFUEL M IP IX W B A P
------- ---- ----  ----- -----  ------ -----  ---- -----  ------ - -- -- - - - -
CLIMB   0.20   -1      0  1500    -1.0   0.0   0.0   0.0  3.0000 1  2 -1 0 0 0 0
CLIMB     -1 0.77   1500 36700    -1.0   0.0   0.0   0.0  3.0000 1  2 -1 0 0 0 1
CRUISE  0.77 0.77     -1 36700  3500.0   0.0   0.0   0.0  1.0000 1  4  0 0 0 0 0
DESCENT 0.77 0.40     -1  1500     0.0   0.0   0.0   0.0  1.0000 1  5  0 0 0 0 0
LOITER  0.40 0.40     -1  1500     0.0  20.0   0.0   0.0  1.0000 1  4  0 0 0 0 0

***** AERODYNAMICS OF THE AEW&C *****
 $ACHAR ABOSB=0.18,  ALMAX=14.0, AMC=40.0, BDNOSE= 0.0, BTEF=0.00,
      CLO=0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
      CLOC=0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
      CLOW=0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
      CMO=0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,
      MACHN=0.75, RALOIT=  0.00, RCLMAX=1.000,
      ROC=.05, ROCAN=.020, SFWF=1.00, SMNDR=0.80,
      SMNSWP=0.0,0.76,0.80,0.82,0.84,0.85,0.86,0.87,0.89,0.91,
      SPANAC= 0.0, SWPMAX=60.0, SWPMIN= 0.0, XCDC=0.60, XCDW=0.60,
      YSWP= 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
      ALELJ=5, INORM=1, ISMNDR=0,
      ISUPCR=0, ITRAP=0, IXCD=1,
      ELLIPC=F, ELLIPH=F, ELLIPW=F, $END
 $AMULT CSF=0.000, ESSF=0.000, FCD=1.000, FCDF=0.975, FCDL=1.0,
      FCDRA=10*1.0,
      FCDO=1.0, FCDW=.500, FCDWB=1.000, FENG=1.000, FINTF=1.000,
      FLBCOR=1.000, FLECOR=1.000, FMDR=0.95, $END
 $ATRIM CAND= 0.0, CFLAP=0.20, CGM=0.25,
      FLDM= 10*1.0,
      FVCAM=10*.75,
      SPANF= 0.75, ZCG= 0.0,
      ITRIM=10*1, IVCAM=1, $END
 $ADET  ALIN= -1.0,0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,
```

```
ALTV= 8*35000.0,
CLINPT=0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
SMN=0.4,0.45,0.5,0.55,0.6,0.65,0.70,0.75,
ICOD=1, IPLOT=1,
ISTRS=1,1,1,1,1,1,1,1,1,1,
ITB=0,0,0,0,0,0,0,0,0,0,
ITS=0,0,0,0,0,0,0,0,0,0,
NALF=10, NMDTL=10, $END
$ADRAG CDBMB= 0.0, 0.0, 0.0, 0.00, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
CDONPT= 0.018203715, 0.018165887, 0.018128696, 0.018092121, 0.018056144, 0.018020747, 0.017985915,
0.017951629, 0.017917874, 0.017884636,
CDSTR=0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
CDTNK= 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
CDEXTR= 0.00, 0.0, 0.0, 0.0, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,
SMNCDO=0.70,0.71,0.72,0.73,0.74,0.75,0.76,0.77,.78,.79
SMNBMB=0.70,0.71,0.72,0.73,0.74,0.75,0.76,0.77,.78,.79
SMSTRS=0.70,0.71,0.72,0.73,0.74,0.75,0.76,0.77,.78,.79
SMTANK=0.70,0.71,0.72,0.73,0.74,0.75,0.76,0.77,.78,.79
SMEXTR=0.70,0.71,0.72,0.73,0.74,0.75,0.76,0.77,.78,.79
ICDO=1, $END
$ATAKE CLLAND=-1.00, CLTO=-1.00, DELFLD=45.0, DELFTO=10.0,
DELLED=30.0, DELLTO=15.0, LDLAND=-1.0, LDTO=-1.0, ALFROT=6.,$END
$APRINT ECHOIN=1, ECHOUT=0, INTM=0, IPBLNT=0, IPCAN=0, IPENG=0, IPEXT=0,
IPFLAP=0, IPFRIC=0, IPINTF=0, IPLIFT=0, IPMIN=0, IPWAVE=0, KERROR=0,
$END
***** GENERAL ELECTRIC CFM56-7B20 TURBOFAN *****
6
$LEWIS
AENDIA=5.08,
AENLE=8.225,  !These values come from reference 15
AENWT=5234.,
BA=5.1,
DIA1=3,
P2P1=28.9,
T3=2900.,
XMACH=0.0,0.6,0.65,0.70,0.75,0.85,
```

```
            ALTD=0.,5*25000.0,
            MACH1=0.85,
            AuxPowerInner=251.0,
 $INLET
  INTYPE=1,
  LM=10.,
  SFPRFP = 1.0,
  NINL  =2,
 $END
 $AFTBD    $END
TRANSPORT
***** AEW&C WEIGHTS *****
 $OPTS   ITAIL=1, WGTO=171000.0, AFMACH=.80, ITHRV = 4, $END
 $FIXW
WELT=21000.,
 $END
 $CMAN
  FEE=0.190,              NNG=1,        RTRTM = 12.,    CTJI = 0.,
  NVEH= 1., 200., 400.,   NDATA=3,       NV=504.,        NFV=1.0,
  RE=65.0,                RT=55.0,       YEAR = 1992.,   API = 0.08,
  LEARNA1=81.5,           LEARNP1=100.,  LEARN1=81.5,    FENGQ =2500.,
  RATE = .667,1.,3.,3.,3.,3.,3.,0.,0.,0., RTRTA = 12.,
  PUNITS = 100.,
  ICONFG=6,     IPROD=1,       IENGS = 1,     IOPS=1,   XFASSY =.05,
  ICSHFLW=1,    IAIRROI=1,      CFENG=1.00,    ENSPAO=.23,  $END
 $COPER
  AMT=0.,       COFL=.10742,    COIL=2.63,     NSL=2,       RCH=500.,
  RESDVL=10.,   RINRST=8.,      DWNPYM=0.,     RL=25.0,     RCSL=3000.,
  DESS=300.,    FINSUR=.35,     U=5000.,       H=32000.,    BDMAIN=200.,
  SL=4600.,4600.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,            ECLIFE=20.,
  FLF=.65,.65,.65,.65,0.,0.,0.,0.,0.,0.,0.,0.,             TXRATE=34.,
  CLF=.65,.65,.65,.65,0.,0.,0.,0.,0.,0.,0.,0.,             GRNDTM =1.,
  DPT=1.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.,0.,
  AI1=.5, AI2=5.2, AI3=215., AI4= 63.55, AI5=2.0, AI6=11., AI7=80.,
  AI8=.0214, AI9=.012, AI10=.0703, AI11=11., AI12=38.,
```

$END

$CMAINT

$END

Appendix B – ACS Output Summary Example

Fuselage Definition (Type 2)

Nose Length.................. 26.892

Nose Fineness Ratio.......... 2.180

Constant Section Length...... 35.509

Afterbody Length............ 43.176

Afterbody Fineness Ratio..... 3.500

Overall Length............... 105.577

Maximum Diameter............ 12.336

Body Planform Area...........1160.470

Passengers

First Class................. 0.

Coach Class................. 115.

Crew........................ 2.

Stewardesses................ 3.

Fuselage Definition

X	R	Area
5.38	2.87	25.82
6.72	3.32	34.59
8.07	3.72	43.53
9.41	4.09	52.45
10.76	4.41	61.19
12.10	4.71	69.62
13.45	4.97	77.63
14.79	5.21	85.12

16.14	5.41	92.01
17.48	5.59	98.24
18.82	5.75	103.75
20.17	5.88	108.49
21.51	5.98	112.42
22.86	6.06	115.51
24.20	6.12	117.73
25.55	6.16	119.07
26.89	6.17	119.52
30.44	6.17	119.52
33.99	6.17	119.52
37.55	6.17	119.52
41.10	6.17	119.52
44.65	6.17	119.52
48.20	6.17	119.52
51.75	6.17	119.52
55.30	6.17	119.52
58.85	6.17	119.52
62.40	6.17	119.52
64.56	6.28	124.00
66.72	6.37	127.64
68.88	6.44	130.38
71.04	6.49	132.19
73.20	6.51	133.04
75.35	6.50	132.90
77.51	6.48	131.74
79.67	6.42	129.55

81.83	6.34	126.32
83.99	6.23	122.05
86.15	6.10	116.74
88.31	5.93	110.39
90.47	5.73	103.03
92.62	5.49	94.66
94.78	5.21	85.31
96.94	4.89	74.99
99.10	4.50	63.72
101.26	4.05	51.45
103.42	3.47	37.92
105.58	2.52	20.01

	Fuselage		Nacelles - 0
Max. Diameter......	12.336	.....	9.675
Fineness Ratio.....	8.559		
Surface Area.......	3693.267	.....	0.000 (each)
Volume............	10665.852		

\

Dimensions of Planar Surfaces (each)

	Wing	H.Tail	V.Tail	Canard	Units
NUMBER OF SURFACES.	1.0	1.0	1.0	1.0	
PLAN AREA..........	1341.0	352.8	284.6	0.0	(SQ.FT.)
SURFACE AREA.......	2701.2	441.5	396.2	0.0	(SQ.FT.)

VOLUME............ 1577.3 144.9 269.8 0.0 (CU.FT.)

SPAN............... 112.571 46.621 23.315 0.000 (FT.)

L.E. SWEEP......... 28.525 34.407 36.036 0.000 (DEG.)

C/4 SWEEP.......... 25.020 30.000 30.000 0.000 (DEG.)

T.E. SWEEP......... 13.299 14.289 7.233 0.000 (DEG.)

ASPECT RATIO 9.450 6.160 1.910 0.000

ROOT CHORD......... 20.556 12.582 19.208 0.000 (FT.)

ROOT THICKNESS..... 29.601 15.099 23.050 0.000 (IN.)

ROOT T/C 0.120 0.100 0.100 0.000

TIP CHORD.......... 3.268 2.554 5.205 0.000 (FT.)

TIP THICKNESS...... 4.707 3.065 6.246 0.000 (IN.)

TIP T/C 0.120 0.100 0.100 0.000

TAPER RATIO 0.159 0.203 0.271 0.000

MEAN AERO CHORD.... 14.003 8.676 13.545 0.000 (FT.)

LE ROOT AT......... 42.371 90.883 83.202 0.000 (FT.)

C/4 ROOT AT........ 47.510 94.029 88.004 0.000 (FT.)

TE ROOT AT......... 62.927 103.466 102.410 0.000 (FT.)

LE M.A.C. AT....... 53.967 97.103 90.061 0.000 (FT.)

C/4 M.A.C. AT...... 57.468 99.272 93.448 0.000 (FT.)

TE M.A.C. AT....... 67.970 105.779 103.607 0.000 (FT.)

Y M.A.C. AT........ 21.336 9.081 0.000 0.000

LE TIP AT.......... 72.963 106.849 100.164 0.000 (FT.)

C/4 TIP AT......... 73.780 107.487 101.465 0.000 (FT.)

TE TIP AT.......... 76.231 109.403 105.369 0.000 (FT.)

ELEVATION.......... -6.168 1.850 3.701 0.000 (FT.)

GEOMETRIC TOTAL VOLUME COEFF 0.786 0.068 0.000

REQUESTED TOTAL VOLUME COEFF 0.786 0.068 0.000

ACTUAL TOTAL VOLUME COEFF 0.786 0.068 0.000

E X T E N S I O N S

	Strake	Rear Extension
Centroid location at.......	0.00	0.00
Area......................	0.00	0.00
Sweep Angle................	0.00	0.00
Wetted Area................	0.00	0.00
Volume....................	0.00	0.00

Total Wing Area........... 1340.97

Total Wetted Area.......... 7232.20

F U E L T A N K S

Tank	Volume	Weight	Density	Fill %
Wing	338.	16923.	50.00	100.0
Fus#1	1029.	51443.	50.00	100.0
Fus#2	0.	0.	50.00	0.0
Total	1367.	68366.		100.0

Mission Fuel Required = 68366. lbs.

Extra Fuel Carrying Capability = 0. lbs.

Available Fuel Volume in Wing = 0. cu.ft.

Mach = 0.65 C.G. Location = 57.5 ft, 0.25 cbar

Altitude = 35000. Reynolds Number per foot = 1.558x10^6

Parasite Drag   Induced Drag

Friction	.0145	Alpha	Cl	Cd	L/D	e	Zone	Cm	Cdtrim	Deltrim	StMrg
Body	.0059	-1.1	-.101	0.0175	-5.8	0.56	2	0.000	0.0000	0.5	0.346
Wing	.0066	0.0	0.000	0.0169	0.0	0.00	2	0.000	0.0000	0.0	0.341
Strakes	.0000	1.1	0.101	0.0175	5.8	0.57	2	0.000	0.0000	-0.5	0.341
H. Tail	.0011	2.2	0.199	0.0192	10.4	0.57	2	0.000	0.0002	-0.9	0.342
V. Tail	.0009	3.3	0.296	0.0221	13.4	0.56	2	0.000	0.0004	-1.4	0.343
Canard	.0000	4.3	0.391	0.0260	15.0	0.56	2	0.000	0.0006	-1.7	0.344
Interference	.0024	5.4	0.484	0.0309	15.6	0.56	2	0.000	0.0009	-2.1	0.346
Base	.0011	6.5	0.576	0.0369	15.6	0.56	2	0.000	0.0013	-2.4	0.352
Wing-Body	.0005	7.6	0.667	0.0440	15.2	0.55	2	0.000	0.0017	-2.8	0.359
Wing-Nacelle	.0000	8.6	0.758	0.0521	14.5	0.55	2	0.000	0.0021	-3.2	0.367

MISSION SUMMARY

PHASE	MACH	ALT	FUEL	TIME	DIST	L/D	THRUST	SFC	Q
=======	====	======	======	=====	======	=====	=======	=====	======
TAKEOFF	0.00	0.	1102.	33.0	4352.9				
CLIMB	0.71	1500.	314.	0.1	1.1	10.74	67673.4	0.612	700.0
CLIMB	0.77	36700.	7063.	6.2	47.1	14.64	20826.0	0.671	191.1
CRUISE	0.77	42365.	49291.	468.9	3451.8	14.90	9158.9	0.617	146.0
DESCENT	0.39	1500.	0.	21.2	118.1	15.02	0.0	0.000	209.0
LOITER	0.40	1500.	1983.	20.0	87.7	14.79	8732.9	0.681	224.4
LANDING				5433.1					

Block Time = 9.159 hr

Block Range = 3618.1 nm

Block Fuel = 59753. Lb

Appendix C – Aerodynamic Code

```
function aero_compare()
x=0;
for z=1:2 %iterate through both the 737-m and 737-c


if z==1
cd ACSOutput
%
batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
input_file = '"C:\Craig\MastersThesis\InputFiles\AWACS\AWACS.acs"';
output_file = '"C:\Craig\MastersThesis\MATLAB\ACSOutput\AWACS\AWACS.out"';
command = sprintf('%s %s %s',batch_file,input_file,output_file);

status = dos(command,'-echo');
cd ..
elseif z==2

batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
input_file = '"C:\Craig\MastersThesis\InputFiles\737_700\737_700.acs"';
output_file =
'"C:\Craig\MastersThesis\MATLAB\ACSOutput\737_700\737_700.out"';
command = sprintf('%s %s %s',batch_file,input_file,output_file);

status = dos(command,'-echo');
cd ..
end

% Block Cdzero
cd C:\Craig\MastersThesis\MATLAB\ACSOutput
fid = fopen('fort.43','r');
CDzero_scan = textscan(fid,'%f %d %f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',2);
fclose(fid);

%% Figures


figure(1)
xlabel('\bf{\alpha}')
ylabel('\bf{C_L}')
title('\bf{C_L vs \alpha}')
hold on
grid on
figure(2)
xlabel('\bf{\alpha}')
ylabel('\bf{C_D}')
title('\bf{C_D vs \alpha}')
hold on
grid on
figure(3)
xlabel('\bf{\alpha}')
ylabel('\bf{M*L/D}')
title('\bf{M*L/D vs \alpha}')
```

```
hold on
grid on
figure(4)
xlabel('\bf{\alpha}')
ylabel('\bf{L/D}')
title('\bf{L/D vs \alpha}')
hold on
grid on
figure(5)
xlabel('C_L')
ylabel('\bf{M*L/D}')
title('\bf{M*L/D vs C_L}')
xlim([0.25 0.70])
ylim([2 16])
hold on
grid on


%% Mach Sweep
% Iterates through the text file pulling the aerodynamics characteristics
% for each mach number.
fid = fopen('fort.43','r');
Mach050_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',12);
fclose(fid);

Mach_sweep{1} = ...
    [Mach050_scan{1} Mach050_scan{2} Mach050_scan{3}...
    Mach050_scan{4} Mach050_scan{5} Mach050_scan{6}...
    Mach050_scan{7} Mach050_scan{8} Mach050_scan{9}];

% Block Mach076
fid = fopen('fort.43','r');
Mach060_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',24);
fclose(fid);

Mach_sweep{2} = ...
    [Mach060_scan{1} Mach060_scan{2} Mach060_scan{3}...
    Mach060_scan{4} Mach060_scan{5} Mach060_scan{6}...
    Mach060_scan{7} Mach060_scan{8} Mach060_scan{9}];

% Block Mach080
fid = fopen('fort.43','r');
Mach070_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',36);
fclose(fid);

Mach_sweep{3} = ...
    [Mach070_scan{1} Mach070_scan{2} Mach070_scan{3}...
    Mach070_scan{4} Mach070_scan{5} Mach070_scan{6}...
    Mach070_scan{7} Mach070_scan{8} Mach070_scan{9}];

% Block Mach082
fid = fopen('fort.43','r');
```

```
Mach074_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',48);
fclose(fid);

Mach_sweep{4} = ...
    [Mach074_scan{1} Mach074_scan{2} Mach074_scan{3}...
    Mach074_scan{4} Mach074_scan{5} Mach074_scan{6}...
    Mach074_scan{7} Mach074_scan{8} Mach074_scan{9}];

% Block Mach084
fid = fopen('fort.43','r');
Mach076_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',60);
fclose(fid);

Mach_sweep{5} = ...
    [Mach076_scan{1} Mach076_scan{2} Mach076_scan{3}...
    Mach076_scan{4} Mach076_scan{5} Mach076_scan{6}...
    Mach076_scan{7} Mach076_scan{8} Mach076_scan{9}];

% Block Mach085
fid = fopen('fort.43','r');
Mach079_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',72);
fclose(fid);

Mach_sweep{6} = ...
    [Mach079_scan{1} Mach079_scan{2} Mach079_scan{3}...
    Mach079_scan{4} Mach079_scan{5} Mach079_scan{6}...
    Mach079_scan{7} Mach079_scan{8} Mach079_scan{9}];

% Block Mach085
fid = fopen('fort.43','r');
Mach080_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',84);
fclose(fid);

Mach_sweep{7} = ...
    [Mach080_scan{1} Mach080_scan{2} Mach080_scan{3}...
    Mach080_scan{4} Mach080_scan{5} Mach080_scan{6}...
    Mach080_scan{7} Mach080_scan{8} Mach080_scan{9}];


% Block Mach085
fid = fopen('fort.43','r');
Mach082_scan = textscan(fid,'%f %f %f %f %f %f %f %f %f',...
    'Delimiter','\t','HeaderLines',96);
fclose(fid);

Mach_sweep{8} = ...
    [Mach082_scan{1} Mach082_scan{2} Mach082_scan{3}...
    Mach082_scan{4} Mach082_scan{5} Mach082_scan{6}...
    Mach082_scan{7} Mach082_scan{8} Mach082_scan{9}];

% Plot figures
```

```
cmap = hsv(numel(Mach_sweep));
Mach_No = [0.4 0.45 0.5 0.55 0.6 0.65 0.7 0.75];
for ii = 1:2:numel(Mach_No)
    figure(1)
   if z==1
       x=1;
   elseif z==2
       x=6;
   end
    figure(1)
    plot(Mach_sweep{ii}(:,1), Mach_sweep{ii}(:,2),'-','Color',cmap(x,:))
    hold on
    legend('M=.40 (AWACS)','M=.50 (AWACS)','M=.60 (AWACS)',...
    'M=.70 (AWACS)','M=.40 (737-700)','M=.50 (737-700)',...
    'M=.60 (737-700)','M=.70 (737-700)','Location','SouthEast')

   figure(2)
    plot(Mach_sweep{ii}(:,1), Mach_sweep{ii}(:,3),'-','Color',cmap(x,:))
    hold on
  figure(3)
    plot(Mach_sweep{ii}(:,1), Mach_No(ii).*Mach_sweep{ii}(:,4),'-
','Color',cmap(x,:))
    hold on
   figure(4)
    plot(Mach_sweep{ii}(:,1), Mach_sweep{ii}(:,4),'-','Color',cmap(x,:))
    hold on
    figure(5)
    hold on
    plot(Mach_sweep{ii}(:,2), Mach_No(ii).*Mach_sweep{ii}(:,4),'-
','Color',cmap(x,:))
    hold on
end
legend('M=.40 (AWACS)','M=.50 (AWACS)','M=.60 (AWACS)',...
    'M=.70 (AWACS)','M=.40 (737-700)','M=.50 (737-700)',...
    'M=.60 (737-700)','M=.70 (737-700)','Location','SouthEast')


End
```

Appendix D – Sample Fuel Consumption Code – PowerFlow to ACS

```
for ii = 1:2

     if ii==1
        batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
        input_file = 'C:\Craig\MastersThesis\InputFiles\AWACS\AWACS.acs';
        output_file =
'C:\Craig\MastersThesis\MATLAB\ACSOutput\AWACS\AWACS.out';
        temp_file = 'C:\Craig\MastersThesis\InputFiles\AWACS\AWACS_temp.acs';
        command = sprintf('%s %s %s',batch_file,input_file,output_file);


load('C:\Craig\MastersThesis\PowerFlow_Mil_v4\PropertyTables\AWACS_Results.ma
t')
        data1 =
reshape(Gen1Power_W.signals.values(1,1,:)/1000,size(Gen1Power_W.signals.value
s(1,1,:),3),1);
        data2 =
reshape(Gen2Power_W.signals.values(1,1,:)/1000,size(Gen2Power_W.signals.value
s(1,1,:),3),1);
        data3 = Tot_Gen_Loss_W.signals.values/1000;
        data4 = LeftPackPower_W.signals.values/1000;
        data5 = RightPackPower_W.signals.values/1000;
        data6 = smooth(sum(hydload.signals.values,2),20);
        data = [data5+data4 data3+data2+data1 data6]';
        datasum = sum(data,1);
        Average_Start_To_Takeoff_Power_Militarized=mean(datasum(1:250));
        Average_Climb_Power_Militarized=mean(datasum(250:600));
        Average_Cruise_Power_Militarized=mean(datasum(603:3663));
        Average_Descent_Power_Militarized=mean(datasum(3664:3980));
        Average_Loiter_Power_Militarized=mean(datasum(3980:4100));
       AUXPOWERINNER = [Average_Start_To_Takeoff_Power_Militarized
Average_Climb_Power_Militarized Average_Cruise_Power_Militarized
Average_Descent_Power_Militarized Average_Loiter_Power_Militarized]; %Power
extracted from Inner turbine (BTU/s)
     else
        batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
        input_file = 'C:\Craig\MastersThesis\InputFiles\737_700\737_700.acs';
        output_file =
'C:\Craig\MastersThesis\MATLAB\ACSOutput\737_700\737_700.out';
        temp_file =
'C:\Craig\MastersThesis\InputFiles\737_700\737_700_temp.acs';
        command = sprintf('%s %s %s',batch_file,input_file,output_file);


load('C:\Craig\MastersThesis\PowerFlow\PropertyTables\CommercialAuxv3.mat')
        data1 =
reshape(Gen1Power_W.signals.values(1,1,:)/1000,size(Gen1Power_W.signals.value
s(1,1,:),3),1);
        data2 =
reshape(Gen2Power_W.signals.values(1,1,:)/1000,size(Gen2Power_W.signals.value
s(1,1,:),3),1);
        data3 = Tot_Gen_Loss_W.signals.values/1000;
```

```
        data4 = LeftPackPower_W.signals.values/1000;
        data5 = RightPackPower_W.signals.values/1000;
        data6 = smooth(sum(hydload.signals.values,2),20);
        data = [data5+data4 data3+data2+data1 data6]';
        datasum = sum(data,1);
        Average_Start_To_Takeoff_Power_Commercial=mean(datasum(1:250));
        Average_Climb_Power_Commercial=mean(datasum(250:600));
        Average_Cruise_Power_Commercial=mean(datasum(603:2323));
        Average_Descent_Power_Commercial=mean(datasum(2323:3040));
        Average_Loiter_Power_Commercial=mean(datasum(3040:3160));
        AUXPOWERINNER = [Average_Start_To_Takeoff_Power_Commercial
Average_Climb_Power_Commercial Average_Cruise_Power_Commercial
Average_Descent_Power_Commercial Average_Loiter_Power_Commercial]; %Power
extracted from Inner turbine (BTU/s)
     end
    for jj = 1:numel(AUXPOWERINNER)

        %% Modify Input Files
        clear A
        cd C:\Craig\MastersThesis\InputFiles
        % Read txt into cell A

        fid = fopen(input_file,'r');
        num = 1;
        tline = fgetl(fid);
        A{num} = tline;
        while ischar(tline)
            num = num+1;
            tline = fgetl(fid);
            A{num} = tline;
        end
        fclose(fid);

        % Change cell A
%         A{131} = sprintf...
%             ('        FCDO=%4.3f,',FCD0(ii));
if ii==1
        A{178} = sprintf...
            ('  AuxPowerInner=%4.1f,',AUXPOWERINNER(jj));
else
           A{174} = sprintf...
            ('  AuxPowerInner=%4.1f,',AUXPOWERINNER(jj));
end

        % Write cell A into txt
        fid = fopen(temp_file, 'w+');
        for num = 1:numel(A)
            if A{num+1} == -1
                fprintf(fid,'%s', A{num});
                break
            else
                fprintf(fid,'%s\n', A{num});
            end
        end
        fclose(fid);
```

```
        movefile(temp_file,input_file)

        %% Run ACS code
      cd C:\Craig\MastersThesis\MATLAB\ACSOutput
status = dos(command,'-echo');
cd ..

        clear A
        %% Read Ouput File
        fid = fopen(output_file,'r');
        num = 1;
        tline = fgetl(fid);
        A{num} = tline;
        while ischar(tline)
            num = num+1;
            tline = fgetl(fid);
            A{num} = tline;
        end
        fclose(fid);

       % tempval = strsplit(A{end-68},'WG');
       % WG(jj) = str2num(tempval{2}(1:10));
        if ii==1
            if jj==1
            tempval = strsplit(A{end-16},' ');
            Fuel_Start_To_Takeoff_AWACS=str2num(tempval{1,5})
            %BFuel(jj) = str2num(tempval{2}(1:end-2));
            elseif jj==2
            tempval = strsplit(A{end-15},' ');
            tempval2 = strsplit(A{end-14},' ');
            Fuel_Climb_AWACS=str2num(tempval{1,5})+str2num(tempval2{1,5})
            elseif jj==3
            tempval = strsplit(A{end-13},' ');
            Fuel_Cruise_AWACS=str2num(tempval{1,5})
            elseif jj==4
            tempval = strsplit(A{end-12},' ');
            Fuel_Descent_AWACS=str2num(tempval{1,5})
            elseif jj==5
            tempval = strsplit(A{end-11},' ');
            Fuel_Loiter_AWACS=str2num(tempval{1,5})
            end


        end

        if ii==2
            if jj==1
            tempval = strsplit(A{end-16},' ');
            Fuel_Start_To_Takeoff_737=str2num(tempval{1,5})
            %BFuel(jj) = str2num(tempval{2}(1:end-2));
            elseif jj==2
            tempval = strsplit(A{end-15},' ');
            tempval2 = strsplit(A{end-14},' ');
            Fuel_Climb_737=str2num(tempval{1,5})+str2num(tempval2{1,5})
            elseif jj==3
            tempval = strsplit(A{end-13},' ');
```

```
            Fuel_Cruise_737=str2num(tempval{1,5})
            elseif jj==4
            tempval = strsplit(A{end-12},' ');
            Fuel_Descent_737=str2num(tempval{1,5})
            elseif jj==5
            tempval = strsplit(A{end-11},' ');
            Fuel_Loiter_737=str2num(tempval{1,5})
            end

        end
    end
 end
             Block_Fuel_AWACS =
Fuel_Start_To_Takeoff_AWACS+Fuel_Climb_AWACS...
  + Fuel_Cruise_AWACS + Fuel_Descent_AWACS+Fuel_Loiter_AWACS

             Block_Fuel_737 = Fuel_Start_To_Takeoff_737 + Fuel_Climb_737 +
...
                 Fuel_Cruise_737 + Fuel_Descent_737 + Fuel_Loiter_737


 x =
[Fuel_Start_To_Takeoff_AWACS/Block_Fuel_AWACS,Fuel_Climb_AWACS/Block_Fuel_AWA
CS...
     Fuel_Cruise_AWACS/Block_Fuel_AWACS,Fuel_Loiter_AWACS/Block_Fuel_AWACS];

  y =
[Fuel_Start_To_Takeoff_737/Block_Fuel_737,Fuel_Climb_737/Block_Fuel_737...
     Fuel_Cruise_737/Block_Fuel_737,Fuel_Loiter_737/Block_Fuel_737];

figure (1)

h = pie(x);
hText = findobj(h,'Type','text'); % text object handles

percentValues = get(hText,'String'); % percent values
str = {'Start Taxi Takeoff: ';'Climb: ';'Cruise: ';'Loiter: '}; % strings
combinedstrings = strcat(str,percentValues); % strings and percent values
h=pie(x,combinedstrings);
hp = findobj(h, 'Type', 'patch');
set(hp(1),'FaceColor',[255/255 51/255 102/255]);
set(hp(2),'FaceColor',[51/255 255/255 204/255]);
set(hp(3),'FaceColor',[51/255 204/255 255/255]);
set(hp(4),'FaceColor',[132/255 112/255 255/255]);

hText = findobj(h,'Type','text'); % text object handles
textPositions_cell = get(hText,{'Position'}); % cell array
textPositions = cell2mat(textPositions_cell); % numeric array
textPositions_cell{1,1}(2)=textPositions_cell{1,1}(2) - .10;
textPositions_cell{1,1}(1)=textPositions_cell{1,1}(1) - .27;
textPositions_cell{4,1}(2)=textPositions_cell{4,1}(2) - .1;
textPositions_cell{4,1}(1)=textPositions_cell{4,1}(1) + .18;
set(hText,{'Position'},textPositions_cell) % set new position
set(hText, 'FontWeight', 'bold', 'FontAngle', 'italic')
```

```
title('737-M Fuel Breakdown', 'FontName', 'avantgarde', 'FontSize', 12, ...
   'FontWeight', 'bold', 'FontAngle', 'italic', 'Color', [.7 0 .7])

figure (2)
v=pie(y);
hText = findobj(v,'Type','text'); % text object handles

percentValues = get(hText,'String'); % percent values
str = {'Start Taxi Takeoff: ';'Climb: ';'Cruise: ';'Loiter: '}; % strings
combinedstrings = strcat(str,percentValues); % strings and percent values
v=pie(y,combinedstrings);
hp = findobj(v, 'Type', 'patch');
set(hp(1),'FaceColor',[255/255 51/255 102/255]);
set(hp(2),'FaceColor',[51/255 255/255 204/255]);
set(hp(3),'FaceColor',[51/255 204/255 255/255]);
set(hp(4),'FaceColor',[132/255 112/255 255/255]);

hText = findobj(v,'Type','text'); % text object handles
textPositions_cell = get(hText,{'Position'}); % cell array
textPositions = cell2mat(textPositions_cell); % numeric array
textPositions_cell{1,1}(2)=textPositions_cell{1,1}(2) - .10;
textPositions_cell{1,1}(1)=textPositions_cell{1,1}(1) - .42;
textPositions_cell{4,1}(2)=textPositions_cell{4,1}(2) - .1;
textPositions_cell{4,1}(1)=textPositions_cell{4,1}(1) + .18;
textPositions_cell{2,1}(2)=textPositions_cell{2,1}(2) - .1;
textPositions_cell{2,1}(1)=textPositions_cell{2,1}(1) - .18;
set(hText,{'Position'},textPositions_cell) % set new position
set(hText, 'FontWeight', 'bold', 'FontAngle', 'italic')

title('737-C Fuel Breakdown', 'FontName', 'avantgarde', 'FontSize', 12, ...
   'FontWeight', 'bold', 'FontAngle', 'italic', 'Color', [.7 0 .7])
```

Appendix E – Aux vs. Fuel vs. Range Code

```
for ii = 1:2
Range=[2900.0 3000.0 3100.0 3200.0 3300.0 3400.0 3500.0];
AUXPOWERINNER = 0:50:300; %Power extracted from Inner turbine (BTU/s)
     if ii==1
        batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
        input_file = 'C:\Craig\MastersThesis\InputFiles\AWACS\AWACS.acs';
        output_file =
'C:\Craig\MastersThesis\MATLAB\ACSOutput\AWACS\AWACS.out';
        temp_file = 'C:\Craig\MastersThesis\InputFiles\AWACS\AWACS_temp.acs';
        command = sprintf('%s %s %s',batch_file,input_file,output_file);
     else
        batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
        input_file = 'C:\Craig\MastersThesis\InputFiles\737_700\737_700.acs';
        output_file =
'C:\Craig\MastersThesis\MATLAB\ACSOutput\737_700\737_700.out';
        temp_file =
'C:\Craig\MastersThesis\InputFiles\737_700\737_700_temp.acs';
        command = sprintf('%s %s %s',batch_file,input_file,output_file);

     end
   for kk=1:numel(Range)
    for jj = 1:numel(AUXPOWERINNER)

        %% Modify Input Files
        clear A
        cd C:\Craig\MastersThesis\InputFiles
        % Read txt into cell A

        fid = fopen(input_file,'r');
        num = 1;
        tline = fgetl(fid);
        A{num} = tline;
        while ischar(tline)
            num = num+1;
            tline = fgetl(fid);
            A{num} = tline;
        end
        fclose(fid);

        if ii==1
        tempval = strsplit(A{113},' ');
        else
        tempval = strsplit(A{105},' ');
        end
        temp=sprintf('%4.1f',Range(kk));

        temp_str=strcat(tempval{1,1},{'  '},tempval{1,2},{'
'},tempval{1,3},...
            {'      '},tempval{1,4},{' '},tempval{1,5},{'  '},temp,{'    '},...
        tempval{1,7},{'   '},tempval{1,8},{'   '}, tempval{1,9},{'
'},tempval{1,10},...
        {' '}, tempval{1,11},{'  '},tempval{1,12},{'  '},tempval{1,13},...
```

```
        {' '}, tempval{1,14},{' '},tempval{1,15},{' '},tempval{1,16},{'
'},tempval{1,17});
        temp_str=char(temp_str);


        if ii==1
        A{113}=temp_str;
        else
         A{105}=temp_str;
        end

if ii==1
        A{178} = sprintf...
            ('  AuxPowerInner=%4.1f,',AUXPOWERINNER(jj));
else
           A{174} = sprintf...
            ('  AuxPowerInner=%4.1f,',AUXPOWERINNER(jj));
end

        % Write cell A into txt
        fid = fopen(temp_file, 'w+');
        for num = 1:numel(A)
            if A{num+1} == -1
                fprintf(fid,'%s', A{num});
                break
            else
                fprintf(fid,'%s\n', A{num});
            end
        end
        fclose(fid);

        movefile(temp_file,input_file)

        %% Run ACS code
      cd C:\Craig\MastersThesis\MATLAB\ACSOutput
status = dos(command,'-echo');
cd ..

        clear A
        %% Read Ouput File
        fid = fopen(output_file,'r');
        num = 1;
        tline = fgetl(fid);
        A{num} = tline;
        while ischar(tline)
            num = num+1;
            tline = fgetl(fid);
            A{num} = tline;
        end
        fclose(fid);

       % tempval = strsplit(A{end-68},'WG');
       % WG(jj) = str2num(tempval{2}(1:10));

        tempval = strsplit(A{end-6},'=');
```

```
        if ii==1
        BFuel(kk,jj) = str2num(tempval{2}(1:end-2));
        else
        BFuel2(kk,jj) = str2num(tempval{2}(1:end-2));
        end

    end
   end

if ii==1
AUXPOWERINNER_kW = AUXPOWERINNER*1.05505585;
figure(1)
surf(Range,AUXPOWERINNER_kW,BFuel');
title('\bf{737-M Range/Fuel Analysis}')
xlabel('\bf{Range [nmi]}');
ylabel('\bf{AuxPowerInner [kW]}');
zlabel('\bf{Block Fuel [lb]}')
hold on

else

AUXPOWERINNER_kW = AUXPOWERINNER*1.05505585;
figure(2)
surf(Range,AUXPOWERINNER_kW,BFuel2');
title('\bf{737-C Range/Fuel Analysis}')
xlabel('\bf{Range [nmi]}');
ylabel('\bf{AuxPowerInner [kW]}');
zlabel('\bf{Block Fuel [lb]}')
end


 end
AUXPOWERINNER_kW = AUXPOWERINNER*1.05505585;
figure(3)
FuelDifference = minus(BFuel,BFuel2);
surf(Range,AUXPOWERINNER_kW,FuelDifference');
title('\bf{Fuel Difference between aircrafts}')
xlabel('\bf{Range [nmi]}');
ylabel('\bf{AuxPowerInner [kW]}');
zlabel('\bf{Block Fuel Difference [lb]}')
```

Appendix F – Aux vs. Fuel Code

```
for ii = 1:2
FCD0 =1.0; % Zero-lift drag coefficient multiplying factor
AUXPOWERINNER = 0:5:235; %Power extracted from Inner turbine (BTU/s)
     if ii==1
        batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
        input_file = 'C:\Craig\MastersThesis\InputFiles\AWACS\AWACS.acs';
        output_file =
'C:\Craig\MastersThesis\MATLAB\ACSOutput\AWACS\AWACS.out';
        temp_file = 'C:\Craig\MastersThesis\InputFiles\AWACS\AWACS_temp.acs';
        command = sprintf('%s %s %s',batch_file,input_file,output_file);
     else
        batch_file = '"C:\Program Files\AVID LLC\ACS Core\runacs.bat"';
        input_file = 'C:\Craig\MastersThesis\InputFiles\737_700\737_700.acs';
        output_file =
'C:\Craig\MastersThesis\MATLAB\ACSOutput\737_700\737_700.out';
        temp_file =
'C:\Craig\MastersThesis\InputFiles\737_700\737_700_temp.acs';
        command = sprintf('%s %s %s',batch_file,input_file,output_file);

     end
    for jj = 1:numel(AUXPOWERINNER)

        %% Modify Input Files
        clear A
        cd C:\Craig\MastersThesis\InputFiles
        % Read txt into cell A

        fid = fopen(input_file,'r');
        num = 1;
        tline = fgetl(fid);
        A{num} = tline;
        while ischar(tline)
            num = num+1;
            tline = fgetl(fid);
            A{num} = tline;
        end
        fclose(fid);

        % Change cell A
%         A{131} = sprintf...
%             ('        FCDO=%4.3f,',FCD0(ii));
if ii==1
        A{181} = sprintf...
            ('  AuxPowerInner=%4.1f,',AUXPOWERINNER(jj));
else
           A{177} = sprintf...
            ('  AuxPowerInner=%4.1f,',AUXPOWERINNER(jj));
end

        % Write cell A into txt
        fid = fopen(temp_file, 'w+');
        for num = 1:numel(A)
```

```
            if A{num+1} == -1
                fprintf(fid,'%s', A{num});
                break
            else
                fprintf(fid,'%s\n', A{num});
            end
        end
        fclose(fid);

        movefile(temp_file,input_file)

        %% Run ACS code
      cd C:\Craig\MastersThesis\MATLAB\ACSOutput
status = dos(command,'-echo');
cd ..

        clear A
        %% Read Ouput File
        fid = fopen(output_file,'r');
        num = 1;
        tline = fgetl(fid);
        A{num} = tline;
        while ischar(tline)
            num = num+1;
            tline = fgetl(fid);
            A{num} = tline;
        end
        fclose(fid);

       % tempval = strsplit(A{end-68},'WG');
       % WG(jj) = str2num(tempval{2}(1:10));

        tempval = strsplit(A{end-6},'=');
        if ii==1
        BFuel_M(jj) = str2num(tempval{2}(1:end-2));
        else
        BFuel_C(jj) = str2num(tempval{2}(1:end-2));
        end

    end
if ii==1
AUXPOWERINNER_kW = AUXPOWERINNER*1.05505585;
figure(1)
scatter(AUXPOWERINNER_kW,BFuel_M)
%xlabel('\bf{Engine Power Load [kW]}');
%ylabel('\bf{Fuel Used [lb]}');
hold on
else

AUXPOWERINNER_kW = AUXPOWERINNER*1.05505585;
figure(1)
scatter(AUXPOWERINNER_kW,BFuel_C,'r')
xlabel('\bf{Engine Power Load [kW]}');
ylabel('\bf{Fuel Used [lb]}');
title('\bf{Fuel Burn vs. Engine Power Draw}');
```

```
legend('737-M','737-C')
end
 end
```